



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

**CARRERA DE INGENIERÍA EN ELECTRÓNICA E
INSTRUMENTACIÓN**

**PROYECTO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO EN ELECTRÓNICA E
INSTRUMENTACIÓN.**

**AUTORES: ALOMOTO IZA, LUIS MEDARDO
QUIMBITA PANCHI, ERICK ALEXANDER**

**TEMA: DISEÑO E IMPLEMENTACIÓN DE CONTROLADORES DIFUSOS
APLICADO A LOS PROCESOS DE PRESIÓN Y FLUJO DEL
LABORATORIO DE REDES INDUSTRIALES Y CONTROL DE
PROCESOS DE LA UNIVERSIDAD DE LAS FUERZAS ARMADAS-ESPE
EXTENSIÓN LATACUNGA, INTEGRADOS POR UNA RED ETHERNET
INDUSTRIAL.**

**DIRECTOR: ING. LLANOS, JACQUELINE
CODIRECTOR: ING. PÉREZ, FABRICIO**

LATACUNGA, ENERO 2015

**UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE
INGENIERÍA EN ELECTRÓNICA E INSTRUMENTACIÓN**

CERTIFICADO

Ing. Llanos, Jacqueline (DIRECTOR)

Ing. Pérez, Fabricio (CODIRECTOR)

CERTIFICAN

Que el trabajo titulado “Diseño e implementación de controladores difusos aplicado a los procesos de Presión y Flujo del laboratorio de redes industriales y control de procesos de la Universidad de las Fuerzas Armadas-ESPE Extensión Latacunga, integrados por una Red Ethernet Industrial”, realizado por Luis Medardo Alomoto Iza y Erick Alexander Quimbita Panchi. Ha sido guiado y revisado periódicamente y cumple normas estatutarias establecidas por la ESPE, en el Reglamento de Estudiantes de la Universidad de las Fuerzas Armadas – ESPE.

Debido a que constituye un trabajo de alto contenido de investigación, que coadyuvará a la aplicación de conocimientos y al desarrollo profesional, si recomiendan su publicación.

El mencionado trabajo consta de un documento empastado y un disco compacto el cual contiene los archivos en forma portátil de Acrobat (pdf).

Autorizan a Luis Medardo Alomoto Iza y Erick Alexander Quimbita Panchi que lo entregue al Ing. Franklin Silva, en su calidad de Director de la Carrera.

Latacunga, Enero del 2015.

Ing. Llanos, Jacqueline
DIRECTOR

Ing. Pérez, Fabricio
CODIRECTOR

**UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE
INGENIERÍA EN ELECTRÓNICA E INSTRUMENTACIÓN**

DECLARACIÓN DE RESPONSABILIDAD

Alomoto Iza, Luis Medardo
Erick Alexander, Quimbita Panchi

DECLARAMOS QUE:

El proyecto de grado denominado “Diseño e implementación de controladores difusos aplicado a los procesos de Presión y Flujo del laboratorio de redes industriales y control de procesos de la Universidad de las Fuerzas Armadas-ESPE Extensión Latacunga, integrados por una Red Ethernet Industrial”, ha sido desarrollado en base a una investigación exhaustiva, respetando derechos intelectuales de terceros, conforme a las citas ubicadas al final de cada párrafo, cuyas fuentes son incluidas en la bibliografía.

Consecuentemente este trabajo es de nuestra autoría.

En virtud de esta declaración, nos responsabilizamos del contenido, veracidad y alcance científico del proyecto de grado en mención.

Latacunga, Enero del 2015

Alomoto Iza, Luis Medardo

C.C. 0503507238

Quimbita Panchi, Erick Alexander

C.C. 0502985575

**UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE
INGENIERÍA EN ELECTRÓNICA E INSTRUMENTACIÓN**

AUTORIZACIÓN

Nosotros, Alomoto Iza, Luis Medardo

Quimbita Panchi, Erick Alexander

Autorizamos a la Universidad de las Fuerzas Armadas–ESPE la publicación, en la biblioteca virtual de la Institución del trabajo titulado “Diseño e implementación de controladores difusos aplicado a los procesos de Presión y Flujo del laboratorio de redes industriales y control de procesos de la Universidad de las Fuerzas Armadas-ESPE Extensión Latacunga, integrados por una Red Ethernet Industrial”, cuyo contenido, ideas y criterios son de nuestra responsabilidad y autoría.

Latacunga, Enero del 2015.

Alomoto Iza, Luis Medardo

C.C. 050350723-8

Quimbita Panchi, Erick Alexander

C.C. 050298557-5

DEDICATORIA

Este proyecto está dedicado con infinito amor a Dios que con sus bendiciones diarias me ha cobijado con ese manto divino y me ha dado fortaleza para seguir adelante día a día dándome salud y vida.

A mis padres, Luis Alfredo y María Soledad quienes han estado ahí con su apoyo incondicional en las buenas y en las malas con su comprensión y cariño a cambio de nada, gracias por ser mi guía y ayudarme a cumplir una de las metas que más e anhelado hasta el día de hoy, todo este esfuerzo va dedicado para ustedes papi y mami los amo.

A mi hermano Eddy y mis hermanas Maribel y Elizabeth por sus deseos y apoyo de superación hacia mí.

A mi novia Erika López por estar siempre apoyándome en cada una de las metas planteadas, gracias por darme esas palabras de aliento cuando más lo necesitaba y principalmente a mi hijita hermosa “Zoe Analía” por ser ese motorcito que me estimula día a día a luchar para ser un hombre de bien.

Luis Alomoto Iza.

El presente proyecto está dedicado de manera muy especial a mis padres Lenin y Janeth, por haberme enseñado a ser un hombre perseverante, y por los grandes sacrificios que han hecho para poder brindarme una educación de calidad, a mi hermana Dayana quien ha estado incondicionalmente a mi lado y a ha sido la motivación más grande para salir adelante.

Erick Quimbita Panchi.

AGRADECIMIENTO

En primer lugar agradezco a mi Dios y a la virgen María por darme un día más de vida y estar siempre ahí protegiéndome y brindándome salud colmándome de muchas bendiciones en todo lugar y en cada momento.

Agradezco enormemente a mis padres por todo el gran apoyo que me han brindado durante toda mi vida hasta el día de hoy, que he logrado cumplir uno de mis más grandes objetivos culminar mi etapa Universitaria, de igual manera agradezco a todos aquellos que me dieron sus consejos de superación que fortalecieron positivamente para hacer de esta meta una realidad.

Eternamente agradecido a la Ingeniera Jacqueline Llanos y al Ingeniero Fabricio Pérez por su gran ayuda en cada una de las dudas que se hicieron presente durante este proyecto, logrando así culminar nuestro proyecto con éxito muchas gracias.

Luis Alomoto Iza.

A Dios que me ha dado la sabiduría, conocimiento, y fortaleza para sobrellevar los altos y bajos de mi vida universitaria.

A mis padres que me brindaron su amor y apoyo incondicional durante toda la carrera, a mi hermana que ha compartido el mayor tiempo a mi lado celebrando en el éxito y consolándome en los tiempos difíciles.

A mi tutora de tesis Ing. Jacqueline Llanos y a mi cotutor Ing. Fabricio Pérez, por haber sido mis guías y apoyarme en la ejecución de este proyecto de graduación; Y a todas aquellas personas que de una u otra forma participaron en la realización de este proyecto.

Erick Quimbita Panchi.

ÍNDICE DE CONTENIDOS

PORTADA	i
CERTIFICADO	ii
DECLARACIÓN DE RESPONSABILIDAD.....	iii
AUTORIZACIÓN.....	iv
DEDICATORIA	v
AGRADECIMIENTO	vi
ÍNDICE DE FIGURAS.....	xi
ÍNDICE DE TABLAS	xv
ÍNDICE DE ECUACIONES	xvi
RESUMEN	xvii
ABSTRACT	xviii
CAPÍTULO 1	1
FUNDAMENTO TEÓRICO	1
1.1. MODELAMIENTO E IDENTIFICACIÓN DE PROCESOS	1
1.1.1. Modelado.....	1
1.1.2. Identificación.....	1
1.1.3. Metodología de la identificación	2
1.2. CONTROLADORES AVANZADOS.....	4
1.2.1. Control Predictivo	4
1.2.2. Control Adaptable o Adaptivo	5
1.2.3. Control Experto.....	5
1.2.4. Control por Lógica Difusa o Control Difuso	6
1.2.5. Control Neuronal	6
1.3. CONTROLADOR DIFUSO	7

1.3.1.	Fundamentos de lógica difusa.....	7
1.3.1.1.	Conjuntos difusos	8
1.3.1.2.	Funciones de pertenencia	11
1.3.1.3.	Operaciones entre conjuntos difusos	12
1.3.1.4.	Fuzzificación.....	15
1.3.1.5.	Base de conocimiento.	16
1.3.1.6.	Inferencia difusa	17
1.3.1.7.	Motor de inferencia.....	18
1.3.1.8.	Defuzzificación	19
1.3.2.	Control difuso	22
1.4.	APLICACIONES DE LOS CONTROLADORES DIFUSOS	25
1.4.1.	Algunas aplicaciones Industriales de lógica difusa	25
1.5.	ETHERNET INDUSTRIAL.....	26
1.6.	CONTROLADORES DE AUTOMATIZACIÓN PROGRAMABLE	28
1.6.1.	NI CompactRIO	30
1.6.2.	NI myRIO.....	30
1.6.2.1.	Las Aplicaciones de NI myRIO:.....	31
1.6.3.	NI Single-Board RIO.....	31
1.6.3.1.	Evaluación NI LabVIEW RIO.....	33
1.7.	SOFTWARE DE DESARROLLO DE SISTEMAS NI LABVIEW.....	34
1.7.1.	Módulo LabVIEW Real-Time	35
1.7.2.	Módulo LabVIEW FPGA.....	36
1.8.	ESTACIONES DE PROCESOS	38
1.8.1.	Estación De Flujo	38
1.8.2.	Estación De Presión	39
1.8.3.	Toolbox Sistema de Identificación de MATLAB	40

1.8.3.1.	Identificación de modelo de datos en matlab.	41
CAPÍTULO 2	45
DISEÑO SIMULACIÓN E IMPLEMENTACIÓN DE LOS ALGORITMOS DE CONTROL	45
2.1.	DISEÑO DEL CONTROLADOR PD DIFUSO MÁS ACCIÓN INTEGRAL DE LA PLANTA DE PRESIÓN.	45
2.2.	DISEÑO DEL CONTROLADOR PID DIFUSO MÁS ACCIÓN INTEGRAL DE LA PLANTA DE FLUJO.	50
2.3.	SIMULACIÓN DE LOS ALGORITMOS DE CONTROL DIFUSOS DE LAS PLANTAS DE PRESIÓN Y FLUJO.	59
2.3.1.	Simulación del Algoritmo de Control de Presión	59
2.3.2.	Simulación del Algoritmo de Control de Flujo.	63
2.4.	SIMULACIÓN A LAZO CERRADO DE LOS ALGORITMOS DE CONTROL DE PRESIÓN Y FLUJO.	67
2.4.1.	Planta de Presión	67
2.4.2.	Planta de Flujo.	68
2.5.	IMPLEMENTACIÓN DE LOS ALGORITMOS DE CONTROL DIFUSOS DE PRESIÓN Y FLUJO.	69
CAPÍTULO 3	74
DISEÑO Y CONFIGURACIÓN DE LA RED ETHERNET	74
3.1.	DISEÑO DE LA RED ETHERNET.	74
3.2.	CONFIGURACIÓN DE LA RED	78
3.3.	HMI DE LA RED ETHERNET	80

CAPÍTULO 4	82
PRUEBAS Y ANÁLISIS DE RESULTADO	82
4.1. PRUEBAS Y ANÁLISIS DE RESULTADO DE LOS CONTROLES IMPLEMENTADOS A LAS ESTACIONES DE PRESIÓN Y FLUJO.....	82
4.1.1 Pruebas y análisis de resultados del control PD difusos más acción integral implementada a la estación de Presión.	82
4.1.2 Pruebas y análisis de resultados del control PID difusos más acción integral implementada a la estación de Flujo.	86
4.2. ANÁLISIS DE RESULTADO DEL CONTROLADOR AVANZADO vs. UN CONTROL TRADICIONAL IMPLEMENTADOS A LAS ESTACIONES DE PRESIÓN Y FLUJO.	90
4.2.1. Resultados del algoritmo PD Difuso más acción Integral vs. El control PID implementado a la estación de Presión.....	90
4.2.2. Resultado del algoritmo PID Difuso más acción Integral vs. El control PID implementado a la estación de Flujo	97
4.3. PRUEBAS DE FUNCIONAMIENTO DE LA RED ETHERNET	104
CAPÍTULO 5	107
CONCLUSIONES Y RECOMENDACIONES	107
5.1 Conclusiones.	107
5.2 Recomendaciones.....	109
REFERENCIAS BIBLIOGRÁFICAS	110

ÍNDICE DE FIGURAS

Figura 1.1. Lógica difusa vs lógica clásica	9
Figura 1.2. Conjuntos difusos para la variable estatura	10
Figura 1.3. Funciones típica de membresía	12
Figura 1.4. Contención de dos conjuntos difusos	13
Figura 1.5. Unión de dos conjuntos difusos	13
Figura 1.6. Intersección de dos conjuntos difusos	14
Figura 1.7. Complemento de dos conjuntos difusos	15
Figura 1.8. Variable lingüística.....	16
Figura 1.9. Variable fuzzificada.....	16
Figura 1.10. Máquina de inferencia difusa	18
Figura 1.11. Método Max	20
Figura 1.12. Método de centroide	20
Figura 1.13. Media de máximos.....	22
Figura 1.14. Controlador difuso.....	23
Figura 1.15. NI CompactRIO	30
Figura 1.16. NI myRIO	31
Figura 1.17. NI Single-Board RIO	32
Figura 1.18. NI LabVIEW RIO Evaluation KIT.....	33
Figura 1.19. Módulo LabVIEW Real-Time	35
Figura 1.20. NI LabVIEW	37
Figura 1.21. Estación de Proceso de Flujo	39
Figura 1.22. Estación de Proceso de Presión	40
Figura 1.23. Sistema de identificación	41
Figura 1.24. Ventana para importar datos al ident.....	42
Figura 1.25. Preprocesamiento de señal	43
Figura 1.26. Estimación y Validación de Modelos.....	44
Figura 2.1.Diagrama de bloque del Control Difuso de la planta de presión	46
Figura 2.2. Funciones de Pertenencia del error de la Planta de Presión	47
Figura 2.3.Funciones de Pertenencia de la derivada del error de la Planta de Presión	47

Figura 2.4. Funciones de Pertenencia para la salida del sistema difuso de la planta de Presión.....	48
Figura 2.5. Diagrama de bloque del Control Difuso de la planta de presión	51
Figura 2.6. Funciones de Pertenencia del error de la Planta de Flujo.	51
Figura 2.7. Funciones de Pertenencia de la Derivada del error de la Planta de Flujo.....	52
Figura 2.8. Funciones de Pertenencia de la Integral del error de la Planta de Flujo.....	53
Figura 2.9. Función de Pertenencia para la salida del sistema difuso para la planta de Flujo.	54
Figura 2.10. Fuzzy en Matlab de la planta Presión	59
Figura 2.11. Conjunto Difuso "Error" de la planta de Presión	60
Figura 2.12. Conjunto Difuso "derivada" de la planta de Presión.....	60
Figura 2.13. Conjunto Difuso "Salida" de la planta de Presión	61
Figura 2.14. Reglas del Controlador de la Planta de Presión.	61
Figura 2.15. Visor de Reglas de la Planta de Presión	62
Figura 2.16. Surface Viewer para la Planta de Presión.....	62
Figura 2.17. Fuzzy en Matlab de la planta de Flujo.....	63
Figura 2.18. Conjunto Difuso "Error" de la planta de Flujo.....	63
Figura 2.19. Conjunto Difuso "derivada" de la planta de Flujo.....	64
Figura 2.20. Conjunto Difuso "integral" de la planta de Flujo	64
Figura 2.21. Conjunto Difuso "Salida" de la planta de Flujo.....	65
Figura 2.22. Reglas Del Controlador de la Planta de Flujo.	65
Figura 2.23. Visor de Reglas de la Planta de Flujo	66
Figura 2.24. Surface Viewer para la Planta de Flujo	66
Figura 2.25. Simulación del Algoritmo de Control Difuso de Presión	67
Figura 2.26. Simulación del Algoritmo de Control Difuso de Flujo	68
Figura 2.27. Nuevo Proyecto FPGA's	69
Figura 2.28. Panel de selección de entradas/salidas digitales y analógicas.	70
Figura 2.29. Programación Real-Time (RT).....	70
Figura 2.30. Programación Control Difuso.....	71
Figura 2.31. Conversor Voltaje-Corriente.....	72
Figura 2.32. Implementación del Control Difuso en la Planta de Presión	72

Figura 2.33. Implementación del Control Difuso en la Planta de Flujo.....	73
Figura 3.1. Puerto Ethernet SB-RIO	75
Figura 3.2. Propiedades de una Variable Compartida	76
Figura 3.3. Variable Compartida.....	77
Figura 3.4. Lectura y escritura en una variable compartida	77
Figura 3.5. Cambio de la Dirección IP de la PC.....	78
Figura 3.6. Cambio de IP de las SB-RIO	79
Figura 3.7. Ping Maestro a Esclavo.....	80
Figura 3.8. HMI de los procesos controlados.....	80
Figura 3.9. Controlador De Presión y Flujo integrados en una red Ethernet.	81
Figura 4.1. SP (20-50 PSI)	83
Figura 4.2. SP (50-20 PSI)	84
Figura 4.3. SP (20-45 PSI)	84
Figura 4.4. SP (45-25 PSI)	85
Figura 4.5. SP (25-35 PSI)	85
Figura 4.6. SP (12-45 inH ₂ O)	87
Figura 4.7. SP (45- 18 in inH ₂ O).....	87
Figura 4.8. SP (18-38 in inH ₂ O).....	88
Figura 4.9. SP (38-28 in inH ₂ O).....	88
Figura 4.10. SP (45-12 in inH ₂ O).....	89
Figura 4.11. SP 20-50 (a) Control Difuso, (b) PID.....	91
Figura 4.12. SP 50-20 (a) Control Difuso, (b) PID.....	92
Figura 4.13. SP 20-45 (a) Control Difuso, (b) PID.....	93
Figura 4.14. SP 45-25 (a) Control Difuso, (b) PID.....	94
Figura 4.15. SP 25-35 (a) Control Difuso, (b) PID.....	95
Figura 4.16. Tiempo de estabilización de los controladores	96
Figura 4.17. Porcentaje de sobreimpulso de los controladores.	96
Figura 4.18. SP 12-45 (a) Control Difuso, (b) PID.....	98
Figura 4.19. SP 45-18 (a) Control Difuso, (b) PID.....	99
Figura 4.20. SP 18-38 (a) Control Difuso, (b) PID.....	100
Figura 4.21. SP 38-28 (a) Control Difuso, (b) PID.....	101
Figura 4.22. SP 45-12 (a) Control Difuso, (b) PID.....	102
Figura 4.23. Tiempo de estabilización de los controladores	103
Figura 4.24. Porcentaje de sobreimpulso	103

Figura 4.25. Dirección IP de la PC.....	104
Figura 4.26. Dirección IP del Dispositivo SB RIO.....	104
Figura 4.27. Envío y recepción de datos en CMD	105
Figura 4.28. VI de la PC Servidor.....	105
Figura 4.29. VI Cliente	106
Figura 4.30. Ventana de visualización de señales	106

ÍNDICE DE TABLAS

Tabla1.1. Características del kit de evaluación NI LabVIEW RIO.....	34
Tabla 2.1. FAM del Controlador de planta de Presión.	49
Tabla 2.2. FAM del Controlador de planta de Flujo.	54
Tabla 4.1. Resultado obtenidos del controlador difuso.	83
Tabla 4.2. Resultado obtenidos del controlador difuso.	86
Tabla 4.3. Control PD Difuso + I vs. Control PID.....	90
Tabla 4.4. Control PID Difuso + I vs. Control PID	97

ÍNDICE DE ECUACIONES

Ec.1.1. Conjunto difuso.....	9
Ec.1.2. Soporte de un conjunto difuso.	11
Ec.1.3. Subconjunto difuso	13
Ec.1.4. Unión de dos conjuntos difusos.	14
Ec.1.5. Intersección de dos conjuntos difusos	14
Ec.1.6. Complemento del conjunto difuso A	15
Ec.1.7. Conjunto difuso salida B'	19
Ec.1.8. Defuzzificación por el método del máximo	19
Ec.1.9. Defuzzificación por el método del centroide o centro de área	20
Ec.1.10. Defuzzificación por el método del centroide salida singleton.....	21
Ec.1.11. Media de máximos	21
Ec.1.12. Salida defuzzificada.....	21
Ec.1.13. Formato de regla mandami.....	23
Ec.4.1. Calcular error.....	82

RESUMEN

En este proyecto de tesis se presenta el diseño, implementación y validación de controladores difusos aplicado a dos procesos con tiempo de respuesta diferente (Presión y Flujo), en el laboratorio de Redes Industriales y Control de Procesos de la Universidad de las Fuerzas Armadas-ESPE extensión Latacunga, integrados por una red Ethernet, para lo cual se utilizó dos NI Single-Board RIO, donde se implementaron los dos algoritmos de control que operan autónomamente.

En la simulación fuera de línea se usa el software Matlab. Para el proceso de Presión se diseña un controlador PD Difuso más una acción integral, la arquitectura del controlador difuso consiste en dos entradas, error y la derivada del error, en donde el error cuenta con 5 conjuntos difusos, la derivada del error con 5 conjuntos difusos y la salida que es un voltaje para el actuador con 7 conjuntos difusos; Se usan 25 reglas y como método de defuzzificación se usa el Centroide. Mientras que para la planta de Flujo se diseña un controlador PID Difuso más una acción integral, la arquitectura del controlador difuso consiste en tres entradas, error, la derivada del error, y la integral del error, donde el error cuenta con 5 conjuntos difusos, la derivada del error cuenta con 5 conjuntos difusos, la integral del error con 5 conjuntos difusos y la salida que es un voltaje para el actuador con 5 conjuntos difusos. Se usan 125 reglas y como método de defuzzificación se usa el Centroide.

Los dos algoritmos de control son analizados con un control tradicional PID, y se nota un mejor desempeño de los controladores propuestos para los dos casos. Además los procesos son integrados por una red que usa protocolo Ethernet, tanto la red como los controladores diseñados fueron validados, verificando su buen funcionamiento.

PALABRAS CLAVES: CONTROLADORES DIFUSOS / DISPOSITIVO DE CONTROL SB-RIO / MÓDULO EN TIEMPO REAL / ALGORITMO DE CONTROL AVANZADO.

ABSTRACT

In this thesis project the design implementation and validation of fuzzy controllers is presented, applied to two processes with different response time (pressure and flow) in the laboratory of Industrial Process and Control Network at the University of the Armed Forces-ESPE Latacunga Extension, integrated over an Ethernet network, for this purpose two NI Single-Board RIO, where the two control algorithms that operate autonomously implemented was used.

In off-line simulation software Matlab is used. For the process of Pressure, fuzzy PD controller is designed more integral action, fuzzy controller architecture consists of two entries, error and error derivative, where the error has 5 fuzzy sets, the derivative of the error with five fuzzy sets and the output is a voltage to the actuator with 7 fuzzy sets; 25 rules are used, the centroid method is used as defuzzification method. While for the plant Flow, fuzzy PID controller is designed more integral action, fuzzy controller architecture consists of three entries, error, the derivative of the error, and the error integral, where the error has 5 fuzzy sets, the derivative of the error with 5 fuzzy sets, the error integral 5 fuzzy sets and the output is a voltage to the actuator 5 fuzzy sets. 125 rules are used, the centroid method is used as defuzzification method.

The two control algorithms are analyzed with a traditional PID control, Have better performance the proposed controllers. Furthermore processes are integrated by a network using Ethernet protocol, both the network and the designed controllers were validated by verifying proper operation.

KEYWORDS: FUZZY CONTROLLERS / CONTROL DEVICE SB-RIO / REAL TIME MODULE / ALGORITHM OF CONTROL ADVANCED.

CAPÍTULO 1

FUNDAMENTO TEÓRICO

1.1. MODELAMIENTO E IDENTIFICACIÓN DE PROCESOS

Es muy normal encontrar en la industria procesos que es necesario regular y del cual no se conoce su función de transferencia, o en general la relación entre la entrada con la salida del proceso.

Podemos encontrar principalmente dos formas mediante las que se puede conseguir un modelo matemático: por modelado o identificación [1].

1.1.1. Modelado

Mediante esta técnica, las ecuaciones del modelo se obtienen a través de la idea de las leyes físicas y químicas que rigen el comportamiento del sistema.

La validez de estos modelos es suficientemente amplio por lo cual son capaces de expresar el comportamiento del proceso en distintos puntos de operación. Los modelos adquiridos suelen ser complicados, y por ende, su mayor utilidad radica en la simulación y el diseño de plantas, mas no en lo concerniente al diseño de sistemas de control. [1]

1.1.2. Identificación

Mediante esta técnica, el modelo se consigue a partir de datos experimentales del proceso sin tener en cuenta las leyes internas, únicamente observando los datos de entrada y salida del sistema.

Esta técnica se emplea generalmente cuando el proceso es medible o controlable. Los experimentos realizados para obtener los valores de las variables del sistema juegan un papel muy importante en la identificación del modelo. Al contrario que el método de modelado, la identificación es útil cuando el objetivo es el diseño de sistemas de control.

1.1.3. Metodología de la identificación

El método de identificación adquiere información del proceso básicamente información de entradas y salidas, de forma directa es posible identificar un proceso adquiriendo los parámetros del modelo identificando su salida esta depende de una variación o cambio de estado en la entrada, se debe aplicar varias perturbaciones en la entrada y de esta manera obtener información suficiente del proceso y poder identificarlo, en sistemas lineales tanto la entrada y su salida tiene la misma frecuencia lo único que varía es su amplitud así como su fase.

El modelo matemático se adquiere a partir de datos experimentales de entradas y salidas del proceso de identificación, estos parámetros hacen referencia a las características y distintivos que tiene el mismo así como: constante de tiempo, ganancia, tiempo muerto entre otros aspectos importantes para el modelamiento matemático.

Los modelos conseguidos de esta manera, poseen la ventaja de que se pueden ajustar al proceso actual, pero al mismo tiempo el nivel de validez es limitado en torno al punto de linealización. [1]

Un proceso de identificación consiste en los siguientes pasos.

1. Diseñar un experimento y seleccionar un conjunto de datos de entrada y salida del proceso, se debe considerar que dependiendo que tan buenos sean los datos colectados el modelo será de la misma manera.
2. Examinar y pulir los datos recolectados filtrando y así remover tendencias y sobresaltos.
3. Seleccionar el modelo apropiado a ser usado con una metodología de identificación específica ya sea: estructuras tipo caja negra, modelamiento físico o modelamiento semi-físico.
4. El criterio de ajuste, la diferencia entre la salida real del proceso con el modelo deben tener una diferencia mínima.
5. Validar el modelo, es importante asegurar que el modelo sea bueno y después proseguir a la aplicación.
6. En caso de que la respuesta no cumpla con las especificaciones requeridas, se debe ir a los pasos anteriores y modificar el método e identificación, la estructura del modelo o los datos experimentales.

Como podemos observar este tipo de metodología en un determinado paso se puede ir a los pasos anteriores para corregir posibles fallas o errores que se hayan encontrado a medida que se implementa este tipo de metodología.

El análisis de datos y la filtración es particularmente de gran importancia, ya que nos permite comparar las características y validar el experimento realizado. El análisis nos ayuda en el uso de funciones de correlación así como el espectro de potencia de las señales involucradas. El filtrado nos ayuda a remover los datos de la información que no se desea que sean reflejadas en el modelo, así como pueden ser el ruidos de alta frecuencia, ruidos de tendencias no deseadas, entre otros.

1.2. CONTROLADORES AVANZADOS

Los controles tradicionales en la actualidad tienen aplicación efectiva en sistemas poco complejos de planta lineales o levemente no-lineales o que no varíen en el tiempo. Lastimosamente en presencia de alguna de estas anomalías o complejidades los esquemas clásicos tienden a ser pocos efectivos y muy poco aplicables.

Con la necesidad de responder o controlar este tipo de procesos, la teoría de control y sistemas ha desarrollado varias soluciones con algorítmicas avanzadas que hoy en día se han implementado en la práctica. Estos desarrollos se denominan Controladores Avanzados.

1.2.1. Control Predictivo

Los controladores predictivos o MPC (Model Predictive Control) se establecen en un algoritmo de optimización que obliga a la planta a seguir una trayectoria definida durante un horizonte de tiempo a futuro.

Estos controladores realizan predicciones futuras, trabajando de manera repetida mediante cálculos en cada uno de sus ciclos analizando el estado de la planta y considerando variaciones y de ser así se cambiara los valores para el proceso y se analizara nuevamente hasta obtener los valores adecuados para un control óptimo del proceso. Por esta razón, este tipo de controladores tienen asociadas poderosas rutinas de identificación. Estos controladores tienen la gran ventaja de manejar plantas multivariables con excesos de grados de libertad [2].

1.2.2. Control Adaptable o Adaptivo

El control adaptable surge como la necesidad de realizar el control de procesos altamente no lineales o con fuertes variaciones en sus dinámicas.

El controlador consiste en un estimador del modelo de la planta que se va readaptando en cada ciclo de control así se tiene en cada intervalo un modelo que es consistente en el punto de operación.

La acción de control se determina a través de inversión del modelo o mediante optimización. El más utilizado es el de tipo lineal por lo que la solución al problema de control es la solución analítica [3].

1.2.3. Control Experto

Es el primer desarrollo de los Sistemas Inteligentes que tratan de emular ciertas características del comportamiento humano tales como pueden ser la toma de decisiones en ambientes complejos, adaptación, tolerancia al ruido, capacidad de aprendizaje.

El controlador es condensado en términos de reglas de acción que se disparan automáticamente dependiendo del estado del sistema. Según la situación puntual de un proceso, el sistema informa al operador de su estado y sugiere la forma de actuar tomando en cuenta que este sistema actúa automáticamente.

El desarrollo es una tarea monótona ya que la extracción del conocimiento se debe realizar mediante entrevistas y revisión de operaciones pasadas, luego debe ser validada por otros expertos. Ya configurada la base del conocimiento esta se efectúa en algún sistema computacional autónomo acoplado en línea al sistema de control central.

Hoy en día estos controladores se utilizan para apoyar la operación de procesos multivariables, diagnóstico de fallas en instrumentación, paradas y puesta en marcha [4].

1.2.4. Control por Lógica Difusa o Control Difuso

El controlador de lógica difusa (Fuzzy Logic) es un tipo de razonamiento que viabiliza llevar al plano cuantitativo hechos descritos en forma vaga o aproximada tales como “Temperatura de horno alrededor de 200°C” o “Válvula medio cerrada”. Y si estas son capaces de ser asociadas mediante reglas de control, entonces su relación puede ser establecida como un Controlador Difuso o “Fuzzy” la misma que tiene la capacidad de entregar valores numéricos a la salida.

Matemáticamente es simple de efectuar por lo que varios controladores apoyados en un microprocesador tienen capacidades de programación de lógica difusa directamente en su hardware acelerando considerablemente la velocidad del ciclo de control.

La veracidad de los sistemas de control difuso radica en la simpleza de sus relaciones, que se basan en un argumento común, que la hacen muy descifrable a todo nivel. Además tiene la ventaja de que se puede emplear en un sistema multivariable, lineal, poco lineal y hasta no-lineales [4].

1.2.5. Control Neuronal

Los controladores neuronales son estructuras matemáticas capaces de asimilar las relaciones entre las entradas y salidas de un proceso. Por ende puede interpretar el comportamiento de un proceso en lazo abierto o directamente la relación en lazo cerrado (controlador neuronal).

Se denomina control neuronal a un lazo de control que posee una red neuronal para operar, ya sea como modelo de planta, controlador o sensor virtual.

Requieren de una gran cantidad de datos para entrenar y validar las variables por lo que deben trabajar en conjunto con sistemas de gestión de información tales como PI System o sistemas de administración de datos del DCS. El entrenamiento radica en un algoritmo de optimización que requiere una gran capacidad de cálculo por lo que su implementación es habitualmente a nivel de software.

Una de las principales causas para utilizar las redes neuronales es el desconocimiento de las relaciones entre las variables de un proceso que impiden la formulación de estrategias más focalizadas [5].

1.3. CONTROLADOR DIFUSO

1.3.1. Fundamentos de lógica difusa

El surgimiento de la teoría de lógica difusa, por Lotfi Zadeh, enfatizo, en que el punto primordial de la lógica difusa es proveer un sistema formal, computacionalmente provisto de conceptos y técnicas para tratar con modos de razonamiento, los cuales son aproximados en lugar de exactos. La lógica difusa está en concepto de grado de pertenencia a un conjunto.

El controlador difuso se han convertido en una de las principales disciplinas matemáticas de gran aplicación hoy en día tales como: lavadoras de ropa, sistema de frenos en trenes, sistemas de aire acondicionado, procesado de imágenes, etc. Todo debido a que son sencillos conceptualmente los sistemas basados en lógica difusa, su disposición para adaptarse a casos particulares, su habilidad para combinar en forma unificada expresiones lingüísticas con datos numéricos, y el no requerir de algoritmos muy sofisticados para su implementación.

Está basado en la relatividad de lo observado proporcionando una manera simple de obtener una conclusión a partir de una información de entrada ambigua. Se adapta fácilmente al mundo real en el que vivimos, incluso puede funcionar con nuestras expresiones, del tipo “hace mucho calor”, “no es muy alto”, “el ritmo del corazón está un poco acelerado”, etc. La clave de esta adaptación al lenguaje se basa en comprender los cuantificadores de nuestro lenguaje, por ejemplo, “mucho”, “medio”, “bajo” y “muy bajo”. Puede entenderse como la posibilidad de asignar más valores de verdad o falsedad a los resultados de falso y verdadero [6].

1.3.1.1. Conjuntos difusos

Una buena estrategia para representar estos conjuntos consiste en recordar algunos aspectos de la teoría de conjuntos clásicos, y a partir de allí hacer una extensión a los conjuntos difusos.

Un conjunto clásico se define como una colección de elementos que existen dentro de un universo, cada uno de los elementos del universo pertenece o no a un determinado conjunto. Por lo tanto, cada conjunto puede definirse completamente por una función de pertenencia, que opera sobre los elementos del universo, y que le asigna un valor de 1 si el elemento pertenece al conjunto y de 0 si no pertenece.

Un conjunto difuso que existe dentro del universo, se define como todos aquellos elementos que pueden ser calificados mediante un grado de pertenencia. Este grado de pertenencia se define mediante la función característica asociada al conjunto difuso: para cada valor que puede tomar la variable x la función característica $\mu_A(x)$ proporciona el grado de pertenencia de ese valor x al conjunto difuso A , definido en la ecuación 1.1:

$$A = \{ x, \mu_A(x) | x \in U \} \quad \text{Ec.1.1}$$

Donde U es el universo de discurso continuo (por ejemplo, todos los números reales: $U = \mathbb{R}$) [11].

Las primeras diferencias que se hacen evidentes entre los conjuntos clásicos y los conjuntos difusos son las siguientes:

- La función de pertenencia asociada a los conjuntos clásicos sólo pueden tener dos valores: 0 ó 1, mientras que en los conjuntos difusos pueden tener cualquier valor entre 0 y 1.
- Un elemento puede pertenecer a un conjunto difuso y simultáneamente pertenecer al complemento de dicho conjunto. Esto no es posible en los conjuntos clásicos, ya que constituiría una violación al principio del tercer excluido.

Visto desde esta perspectiva se puede considerar que la lógica clásica es un caso límite de la lógica difusa. Así pues los conjuntos difusos pueden ser considerados como una generalización de los conjuntos clásicos, como se representa en la Figura 1.1 [7].

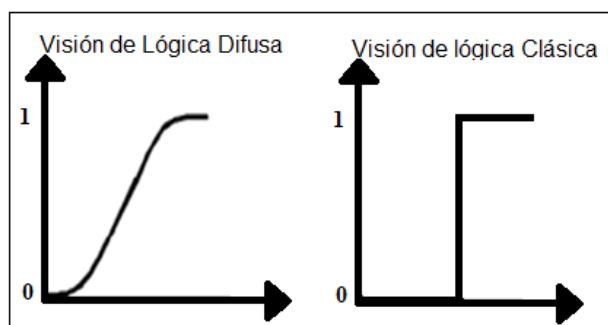


Figura 1.1. Lógica difusa vs lógica clásica. [7]

Con la implementación de un controlador difuso se permite tratar información imprecisa, como: estatura alta, media o baja de una persona. Así, por ejemplo en la Figura 1.2, se representa a un individuo es bajo si se

tiene una estatura inferior a 1.60 metros, un individuo es mediano, si tiene una estatura superior a 1.60 metros e inferior a 1.80 metros y un individuo es alto, si tiene una estatura superior o igual a 1.80 metros, con lo que se lograría una clasificación en conjuntos clásicos. Sin embargo, qué tan grande es la diferencia que existe entre dos individuos, uno con estatura de 1.79 metros y otro de 1.80 metros; este centímetro de diferencia quizás no represente en la práctica algo significativo, y a pesar de eso los dos individuos han quedado rotulados con etiquetas distintas: uno es mediano y el otro es alto. Si se optase por efectuar la misma clasificación con conjuntos difusos estos cambios abruptos se evitarían, debidos a que las fronteras entre los conjuntos permitirían cambios graduales en la clasificación [7].

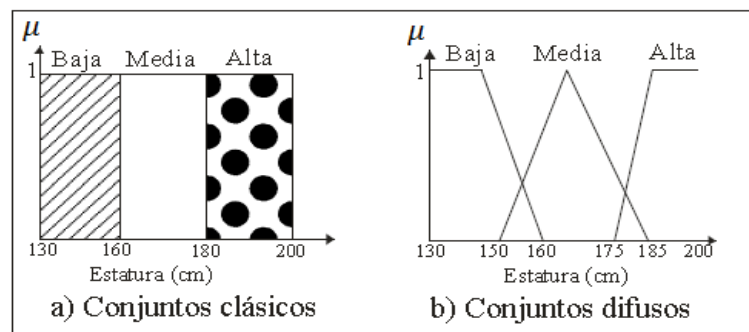


Figura 1. 2. Conjuntos difusos para la variable estatura. [7]

Conceptos importantes de conjuntos difusos:

- **Etiqueta.** Nombre descriptivo usado para identificar un conjunto difuso.
- **Variable lingüística.** Es aquella que puede tomar por valor términos del lenguaje natural, como: mucho, poco, positivo, negativo, etc., que son los calificativos que desempeñan el papel de etiquetas en un conjunto borroso.

- **Universo de discurso.** Rango de todos los valores posibles aplicados a una variable lingüística.

1.3.1.2. Funciones de pertenencia

Se refiere a una función característica asociada al conjunto difuso, que también puede ser nombrada como función de membresía o pertenencia. Así, si se define un conjunto difuso “A” con n elementos: la función de membresía quedaría definida por $\mu_A(x)$, donde para cada valor que pueda tomar un elemento o variable de entrada x , la función de membresía $\mu_A(x)$ proporciona el grado de pertenencia de este valor de x al conjunto difuso A. La única condición que deben de cumplir es que $\mu_A(x) \in [0,1]$ [8].

Conceptos exclusivos e inherentes a la teoría de conjuntos difusos:

- El soporte de un conjunto difuso A en el universo de discurso U es un conjunto “crisp” (numérico) que contiene todos los elementos de U que tienen un valor de pertenencia mayor a cero en A, ilustrado en la ecuación 1.2:

$$\text{sop}(x) = \{x \in U \mid \mu_A(x) > 0\} \quad \text{Ec.1. 2.}$$

Si el soporte de un conjunto difuso no contiene ningún elemento, se tiene un conjunto difuso vacío. Si el soporte de un conjunto difuso es solo un punto, se tiene lo que se conoce como “singleton” difuso.

- El punto de cruce de un conjunto difuso, es el punto donde $\mu_A(x) = 0,5$.
- Dos conjuntos difusos A y B son iguales si y sólo si sus funciones de pertenencia $\mu_A(x)$ y $\mu_B(x)$ son iguales.

La forma de utilizar la función de membresía, depende del criterio aplicado en la resolución de cada problema y varía en la experiencia del usuario. Existen diferentes tipos de funciones de membresía como: singleton, triangular, trapezoidal, pi, monotónica, S, gama, gaussiana o forma de campana, y sigmoïdal, pero las más utilizadas por su simplicidad matemática y operación son: triangular y trapezoidal, como se muestra en la Figura 1. 3.

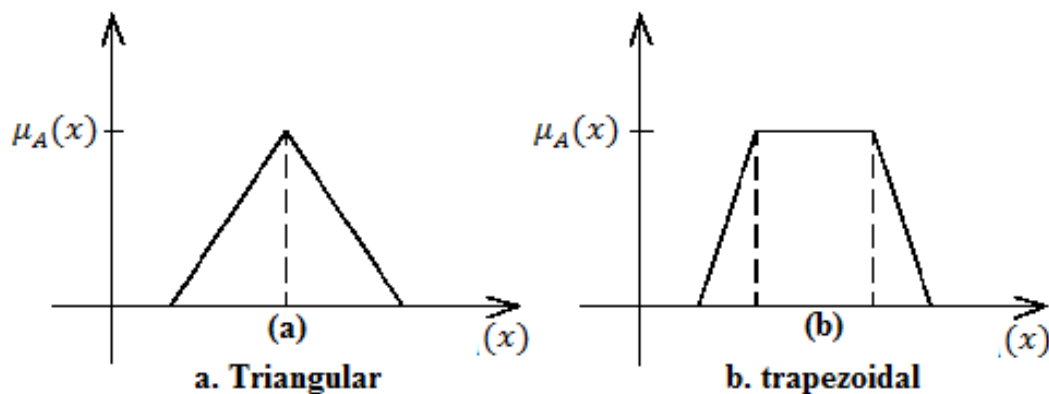


Figura 1. 3. Funciones típica de membresía. [8]

El número de funciones de membresía depende del diseñador. A mayor número de funciones tendremos mayor resolución, pero también mayor complejidad computacional, por lo que se recomienda un número impar, de 3 a 7, con un solapamiento del 20% al 50%.

1.3.1.3. Operaciones entre conjuntos difusos

Las operaciones entre conjuntos difusos utilizadas con más frecuencia son: unión, intersección y complemento [9].

Subconjunto o contención.

Para dos conjuntos difusos A y B sobre el mismo universo de discurso, y con funciones de pertenencia $\mu_A(x)$ y $\mu_B(x)$, se dice que el conjunto difuso A está contenido en el conjunto difuso B, si y solamente

si $\mu_A(x) \leq \mu_B(x)$. Es decir el subconjunto difuso está definido como se observa en la ecuación 1.3 e ilustrada en la Figura 1. 4:

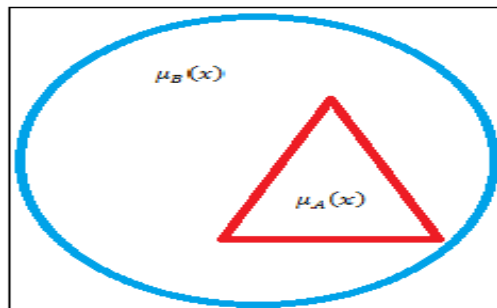


Figura 1. 4. Contención de dos conjuntos difusos. [9]

$$A \subseteq B \Leftrightarrow \mu_A(x) \leq \mu_B(x) \quad \text{Ec.1. 3}$$

Unión

Para dos conjuntos difusos A y B sobre el mismo universo de discurso, y con funciones de pertenencia $\mu_A(x)$ y $\mu_B(x)$ se dice que el resultado de efectuar la operación de unión entre estos dos conjuntos (que en lógica binaria es el equivalente de una operación OR), es un nuevo conjunto difuso $A \cup B$ definido sobre el mismo universo, y con función de pertenencia $\mu_{A \cup B}(x)$, como se representa en la Figura 1. 5, definida por la ecuación 1.4:

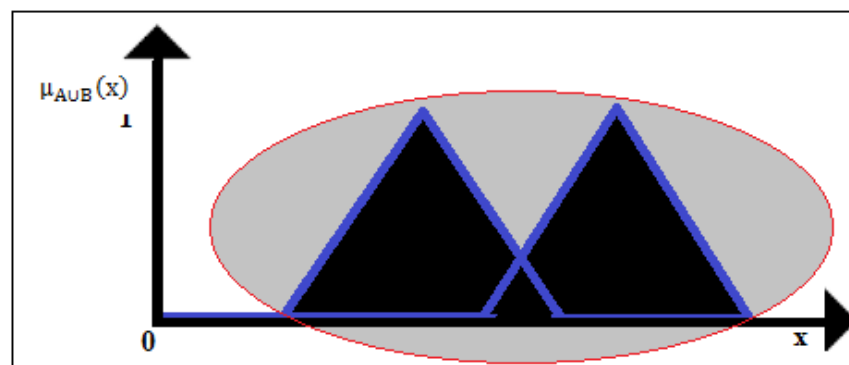


Figura 1. 5. Unión de dos conjuntos difusos. [9]

$$\mu_{A \cup B}(x) = \max[\mu_A(x), \mu_B(x)]$$

Ec.1. 4.

Intersección

Para dos conjuntos difusos A y B definidos sobre el mismo universo de discurso, y con funciones de pertenencia $\mu_A(x)$ y $\mu_B(x)$, se dice que el resultado de efectuar la operación de intersección entre estos dos conjuntos (que en lógica binaria es el equivalente AND), es un nuevo conjunto difuso $A \cap B$ definido sobre el mismo universo, y con función de pertenencia $\mu_{A \cap B}(x)$ definida por la ecuación 1.5 e ilustrada en la Figura 1. 6:

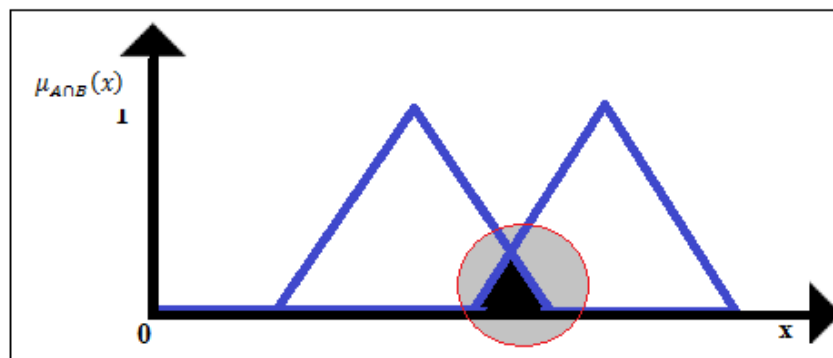


Figura 1. 6. Intersección de dos conjuntos difusos. [9]

$$\mu_{A \cap B}(x) = \min[\mu_A(x), \mu_B(x)]$$

Ec.1. 5.

Complemento

Para un conjunto difuso A definido sobre un universo de discurso U, y cuya función de pertenencia es $\mu_A(x)$, se dice que el resultado de efectuar la operación de complemento (que en lógica binaria es el equivalente de la operación NOT), es un nuevo conjunto difuso A' representada en la Figura 1. 7, definido sobre el mismo universo, y con función de pertenencia $\mu_{A'}(x)$, definida por la ecuación 1.6:

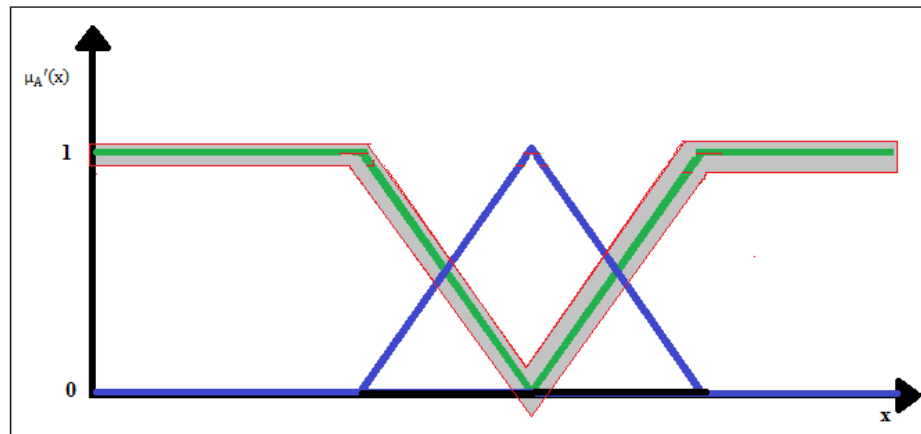


Figura 1. 7. Complemento de dos conjuntos difusos. [9]

$$\mu_{A'}(x) = 1 - \mu_A(x)$$

Ec.1. 6

1.3.1.4. Fuzzificación

En esta etapa se transforman los parámetros físicos de un sistema en variables difusas, donde a cada variable física se le asigna un grado de pertenencia en cada uno de los conjuntos difusos que se han considerado, mediante las funciones de membresía asociadas a estos conjuntos difusos.

El primer paso en la fuzzificación es dividir el universo de discurso, asignando etiquetas en cada variable difusa. Después, se establecen las funciones de membresía para dar significado numérico a cada etiqueta. En el proceso, las funciones de membresía definidas sobre las variables de entrada mapean a los datos presentes para determinar el grado de pertenencia.

Por ejemplo, en la Figura 1. 8, se muestra los valores difusos de la variable lingüística o variable difusa temperatura podrían ser MUY BAJO, BAJO, MEDIO, ALTO, MUY ALTO.

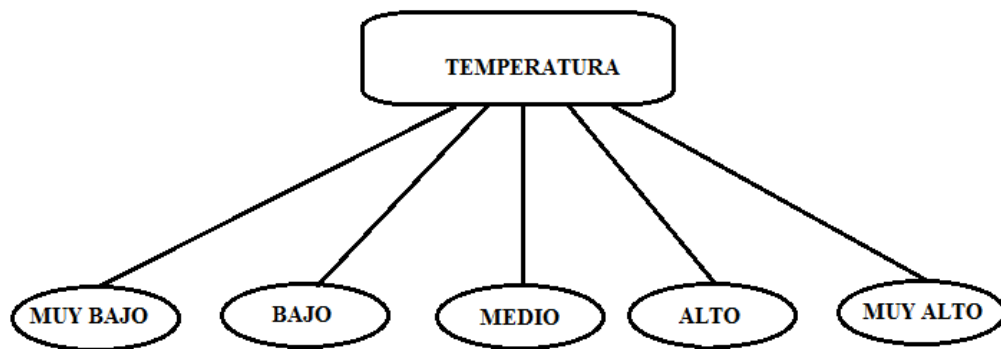


Figura 1. 8. Variable lingüística. [9]

Estos argumentos se conocen como valores difusos, cada uno de ellos está modelado por su propia función de pertenencia como es representada en la Figura 1. 9.

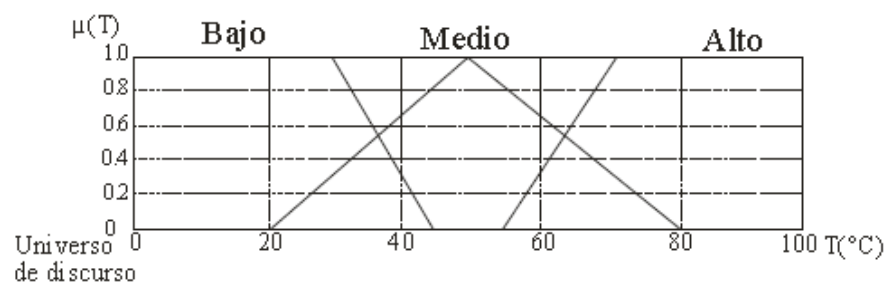


Figura 1. 9. Variable fuzzificada. [9]

1.3.1.5. Base de conocimiento.

Esta base del conocimiento de sistemas difusos consisten de dos componentes: una base de datos y una base de reglas del comportamiento del sistema. Las bases de datos contienen la definición lingüística de las variables tanto de entrada como de salida. Define la semántica de cada variable lingüística.

Las bases de reglas combinan uno o más conjuntos difusos de entrada, llamados antecedentes o premisas, y les asocian un conjunto difuso de salida, llamado consecuente o consecuencia [9].

If premisa(antecedente), THEN conclusión(consecuente)

Los conjuntos difusos de la premisa se asocian mediante conjuntivas lógicas como (AND) y (OR). Esto permite expresar el conocimiento del que se dispone sobre la relación entre antecedentes y consecuentes mediante implicaciones difusas. Para expresar este conocimiento de forma completa, normalmente se precisa de varias reglas que se agrupan como una tabla de reglas o bien como una memoria asociativa difusa o FAM. Las FAM son matrices que representan la consecuencia de cada regla definida para cada combinación de dos entradas.

Existen diversas formas de derivar las reglas, entre las que destacan las basadas en:

- **La experiencia de expertos y el conocimiento de ingeniería de control.** La base de reglas se determina a partir de observar y analizar la forma en que el operador maneja el sistema.
- **La modelación del proceso.** Los parámetros de la base de conocimiento se obtienen a partir de datos de entrada y salida del proceso.

1.3.1.6. Inferencia difusa

Se le denomina inferencia difusa a la manera de obtener conjuntos difusos en base a combinaciones de otros conjuntos difusos mediante reglas **IF (antecedente).... THEN (consecuencia)**.

Existen dos métodos básicos de inferencia entre reglas o leyes de inferencia, modus ponens generalizado (GMP) y el modus tolens generalizado (GMT), por ende son las más utilizadas debido a su simplicidad para obtener conclusiones simples por ejemplo:

Modus ponens:

<i>Regla : si A es cierto</i>	}	<i>B es cierto</i>
ENTONCES		
<i>B es cierto</i>		
<i>Hecho : A es cierto</i>		

Modus Tollens:

<i>Regla : si A es cierto</i>	}	<i>A es falso</i>
ENTONCES		
<i>B es cierto</i>		
<i>Hecho : B es falso</i>		

Para llevar a cabo la inferencia, se utilizan dispositivos de inferencia difusa, estos interpretan las reglas de tipo If-then de una base de reglas, con el fin de obtener los valores de salida a partir de los actuales valores de las variables lingüísticas de entrada al sistema [6].

1.3.1.7. Motor de inferencia

La función del motor de inferencia se basa en dos aspectos importantes los cuales son analizar las reglas, de ser necesario aumentar nuevas y decidir el orden de ejecución de la inferencia difusa, en la Figura 1. 10, se indica la secuencia de cálculos que realiza el motor de inferencia [6][8] incluye:

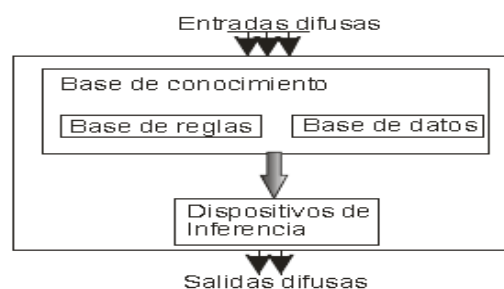


Figura 1. 10. Máquina de inferencia difusa. [8]

- Determina el grado de cumplimiento de cada regla a partir de los grados de pertenencia de las variables de entrada obtenidas en la fuzzificación.
- Utiliza un dispositivo de inferencia para interpretar la base de reglas y de datos.
- Genera una salida difusa, que es un conjunto difuso salida B', siendo éste la unión de los m conjuntos difusos mostrado en la ecuación 1.7.

$$\mu_{B'}(y) = \mu_{B^1}(y) + \dots + \mu_{B^m}(y) \quad \text{Ec.1. 7}$$

1.3.1.8. Defuzzificación

Es el proceso inverso a la fuzzificación el cual transforma un conjunto difuso conformado por variables lingüísticas con sus concernientes grados de pertenencia, en un valor real es decir un número.

Para realizar esta tarea se utilizan diversos métodos:

Sea B' un conjunto difuso definido sobre un universo de discurso U, y cuya función de pertenencia del conjunto de salida B' es $\mu_{B'}(y)$.

- **Defuzzificación por el método del máximo (CoM).** Es el caso en donde la función característica del conjunto difuso tiene su máximo valor, tomando ese valor como salida, definida como en la ecuación 1.8, y representada en la Figura 1. 11. [10]

$$\mu_{B'}(y') \geq \mu_{B'}(y) \quad \text{Ec.1. 8}$$

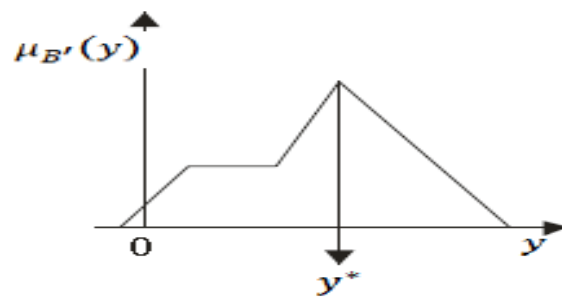


Figura 1. 11. Método Max. [10]

- **Defuzzificación por el método del centroide o centro de área (CoG).** Se calcula el centro de gravedad del área limitada por la curva de la función de membresía como se define en la ecuación 1.9, representa en la Figura 1. 12.

$$y^* = \frac{\int \mu_{B'}(y) * y dy}{\int \mu_{B'}(y) dy} \quad \text{Ec.1. 9}$$

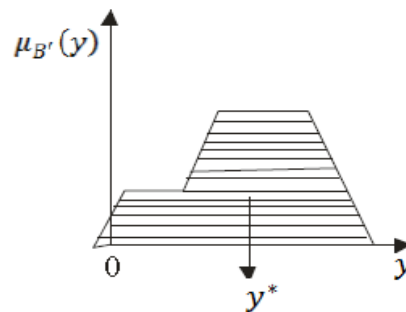


Figura 1. 12. Método de centroide. [10]

Dónde:

- $\int \mu_{B'}(y) dy$: es el área de la ejecución difusa $\mu_{B'}$.
- y^* : es el centro de gravedad (acción de control concreta).

La utilización de este método implica un gran esfuerzo computacional debido a la necesidad de calcular el área de la función de membresía $\mu_{B'}(y)$, se puede simplificar calculando el centro de gravedad global a partir de los centros de gravedad locales de cada conjunto difuso y^* .

El método se reduce cuando los valores del conjunto difuso de la variable lingüística de salida son singletons. Dado que un singleton es un conjunto difuso cuya función de membresía es siempre igual a 1 y cuyo soporte es un sólo punto en μ , el área I_i del respectivo singleton es siempre igual a 1; por otro lado, el centroide del singleton es el mismo punto de μ que lo define en la ecuación 1.10.

$$y^* = \frac{\sum_{i=1}^n W_i S_i}{\sum_{i=1}^n W_i} \quad \text{Ec.1. 10}$$

Dónde:

- y^* = centro de gravedad global (acción de control concreta).
- w_i = grado de influencia de la regla i.
- s_i = singleton en la ejecución de la regla i.
- $i = 1, 2, \dots, n$.
- n = número de reglas.

- **Media de máximos (MoM).** La salida está definida por la ecuación 1.11, en donde \bar{y} es el valor medio de los puntos de máximo grado de pertenencia de la función de membresía [11].

$$y^* = \frac{\sum_t \bar{y}_t}{n} \quad \text{Ec.1. 11}$$

Por ejemplo, en la Figura 1. 13, se representa la salida defuzzificada y la salida está definida por la ecuación 1.12.

$$y^* = \frac{a+b}{2}. \quad \text{Ec.1. 12}$$

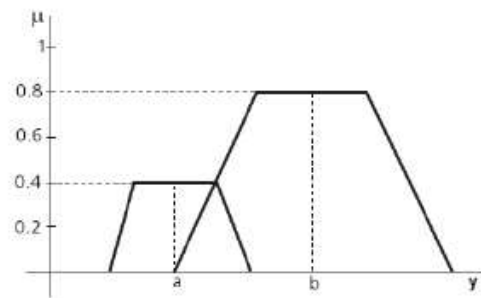


Figura 1. 13. Media de máximos. [11]

1.3.2. Control difuso

El control de sistemas utilizando lógica difusa es una de las aplicaciones más interesantes de la teoría difusa, en el diseño de sistemas de control, a partir de unas entradas se deben generar unas salidas para actuar sobre determinados mecanismos.

La lógica difusa comenzó a utilizarse en controladores por su simplicidad, ya que no requiere de operaciones matemáticas complejas, permitiendo cambiar el diseño mediante la descripción del funcionamiento con lenguaje natural.

Los controladores difusos simples por lo general se representan como se muestra en la Figura 1. 14, en donde sus partes interactúan de la siguiente manera: el bloque de interfaz de fuzzificación se encarga de convertir las entradas R (referencia) y señal del sensor en conjuntos difusos, apoyado por el módulo de base de conocimiento, el cual contiene el conocimiento sobre las entradas y salidas difusas; las entradas fuzzificadas son entregadas al bloque de motor de inferencia, este bloque, apoyado en un conjunto de reglas de la forma *If...then...* almacenadas en la base de reglas, produce varios conjuntos difusos para que el bloque de interfaz de defuzzificación los tome y los convierta en salidas no difusas, aptas para la planta (A) [10].

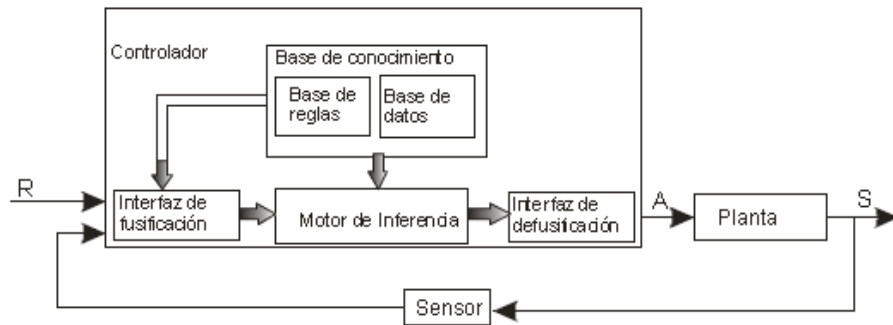


Figura 1. 14. Controlador difuso. [10]

Los controladores difusos se pueden clasificar basados en sus reglas de inferencia en dos tipos: Controlador tipo Takagi-Sugeno y Mamdani [5].

El controlador Mamdani sigue el siguiente formato de reglas que se muestra en la definición:

Sean A_1^l, A_n^l y B^l conjuntos difusos sobre un universo de discurso U , y $x = (x_1 \dots x_n)^T \in U_1 \times \dots \times U_n$ son las entradas, mientras que y es la salida del sistema difuso. Si llamamos M al número de reglas If-THEN de la base, entonces está definida como se muestra en la ecuación 1.13.

$$\text{If } x_1 \text{ es } A_1^l \text{ and } x_2 \text{ es } A_2^l \text{ and } \dots \text{ and } x_n \text{ es } A_n^l \text{ THEN } y^l = f^l(x) \quad \text{Ec.1. 13}$$

Los sistemas difusos con n entradas y una sola salida, se conocen como MISO (Multiple Input Single Output), mientras que los que tienen varias salidas se conocen como MIMO (Multiple Input Multiple Output) [6].

La arquitectura del control a utilizar depende de la aplicación concreta. No resulta fácil realizar una clasificación genérica de todas las arquitecturas posibles de controladores basados en lógica difusa, no obstante, consideraremos los siguientes tres grandes grupos de controladores.

- **Controladores difusos directos sin optimización.** La estructura típica consiste en un preprocesado de las variables de entrada y un postprocesado de la salida del controlador, con el fin de adecuarla al

proceso que se ha de controlar El tipo de preprocesado de las entradas define la clase de controlador, en este sentido, los tipos más usuales de controlador son:

1. Controlador proporcional: $X = f(e)$
2. Controlador integral: $X = f(s)$
3. Proporcional-derivativo: $X = f(e, \Delta e)$
4. Proporcional-integral: $X = f(e, s)$
5. Con realimentación no lineal: $X = f(R, S)$

En las expresiones anteriores se denota con e el error, con Δe su derivada y con s su integral. R Representa las variables de control y S las salidas.

Estos controladores permiten realizar control de sistemas utilizando una descripción lingüística de las reglas de control. Estas reglas han de obtenerse del conocimiento que disponen los expertos sobre el control del sistema, o bien por procedimientos heurísticos.

- **Controladores difusos directos con optimización.** Éstos parten de la estructura de los controladores difusos directos sin optimización, añadiendo elementos que permiten ajustar sus parámetros internos con el fin de mejorar su eficiencia. Según el tipo de ajuste se distinguen diversos tipos de optimizaciones:

- a) Controladores difusos auto-organizados
- b) Controladores difusos con auto-aprendizaje
- c) Controladores basados en modelado difuso

- **Controladores difusos híbridos.** Son aquellos controladores formados por dos controladores interconectados, de los cuales uno es convencional (como los PID) y el otro es difuso. El primero se

encarga básicamente del control, garantizando un comportamiento estable, mientras que el controlador difuso actúa en paralelo, introduciendo el componente heurístico en el proceso. Este segundo controlador difuso, también puede emplearse para el ajuste de los parámetros del controlador convencional.

1.4. APLICACIONES DE LOS CONTROLADORES DIFUSOS

Las Plantas modernas en la industria exhiben un comportamiento fuertemente alineal, y los nuevos métodos de producción son bastante complejos para poder modernizarlos adecuadamente. Por tal motivo, técnicas desarrolladas décadas atrás han resurgido como métodos eficientes para diseñar, optimizar y mantener sistemas altamente complejos. Tal es el caso de los sistemas de Control Borroso o difuso y Neuroborroso o neurodifuso. Hoy en día el término “Fuzzy Logic” se encuentra presente en muchos artefactos hogareños.

1.4.1. Algunas aplicaciones Industriales de lógica difusa:

Las aplicaciones industriales en este mundo moderno son de gran importancia para el desarrollo de los procesos industriales ya sea para el control, monitoreo y automatización de los diferentes procesos y plantas de control existentes en las industrias de baja y alta escala, algunas de las aplicaciones podemos manifestar a continuación:

- **Ahorro de Energía:** Control de Lavarropas. Control adaptativo de Sistemas de Calefacción.
- **Industria Automotriz:** Sistema de Frenos Antibloqueo. Control de Motores de Combustión Interna. Control de Crucero. Control de Aire Acondicionado para Automóviles.
- **Automación Industrial:** Control de Decantador. Control de Producción de Plásticos. Control de Reciclado de Agua. Control

de Horno de Cocción de bizcochos.

- **Análisis de Datos:** Aplicaciones Médicas. Aplicaciones Sensoriales.
- **Otras Aplicaciones:** Cargador de Baterías. Estabilización de Imágenes
- **Digitales en Cámaras.** Control de Anestesia.

1.5. ETHERNET INDUSTRIAL

Sin duda alguna, Ethernet se ha transformado en el estándar de la conectividad para ambientes corporativos y hogareños por su alta velocidad, bajo costo, facilidad de instalación y mantenimiento, entre otros factores. Hace algunos años, comenzó a popularizarse el concepto de Ethernet Industrial, que engloba el uso de la tecnología Ethernet para aplicaciones de control y automatización en un ambiente industrial. Sin embargo, el usuario debe preocuparse por algunas prestaciones de este protocolo para su uso exitoso en el piso de planta.[13]

- **Determinismo**

Tal como el movimiento Ethernet Industrial continúa en la base de fabricación, un elemento clave de preocupación es el rendimiento de extremo a extremo.

Por esto, el determinismo, es decir, la capacidad de garantizar que un paquete es enviado y recibido en un determinado período de tiempo, es un importante objetivo para el diseño de las redes industriales. Las pruebas de rendimiento de redes conmutadas (switched) y enrutadas (routing) han demostrado que es posible proporcionar comunicación en tiempo real en el dominio de la red, utilizando la calidad de servicio.

Determinismo para la información de control crítico es guardado a

través del uso de la capa 2 VLANs para IEEE802.1p/1Q y TOS/DiffServ para la capa 3.

- **Redundancia**

Las redes Ethernet Industrial deben ser altamente confiables y seguir en funcionamiento durante duras condiciones ambientales, interrupciones accidentales de red y fallas de los equipos. La caída de una red puede ser peligrosa y cara. La confiabilidad de la red es en gran medida conseguida por el uso de Redundancia para todos los vínculos críticos. Hay cuatro esquemas de redundancia populares para Ethernet: Spanning Tree Protocol (STP), Rapid Spanning Tree Protocol (RSTP), Link Aggregation (Trunking) y topología de anillos propietaria.

- **Seguridad**

Si bien la integración creciente de tecnologías de la información e Ethernet Industrial presenta el potencial para ofrecer nuevos niveles de beneficio en las operaciones industriales, también plantea posibles vulnerabilidades. El acto de monitoreo y el análisis de datos procedentes de sistemas de control en niveles de planta, significa que el equipo también se extiende en la otra dirección. Esto aumenta enormemente la exposición de la ampliación de la red de intrusiones y amenazas. Factores internos ofrecen diferentes riesgos. Por ejemplo, cuando la red está sobrecargada debido a los dispositivos defectuosos o errores de funcionamiento, los switches y routers pueden ofrecer un pequeño alivio. Ethernet Industrial puede usar muchos métodos para garantizar la confidencialidad e integridad de la red.

Estas medidas de seguridad de red se pueden agrupar en varias categorías, incluidas las de control de acceso y autenticación, seguridad en la conectividad y administración.

- **Monitoreo**

Los puertos espejo (Mirror port) proporcionan a los ingenieros y técnicos herramientas de monitoreo en tiempo real para el comportamiento del sistema. El Monitoreo permite observar dentro de la operación activa de la red para tipos y cantidades de tráfico esperado. Igualmente importante es la identificación del uso inesperado de la red, para identificar fugas de la empresa a las redes de la fábrica.

El uso de puerto espejo en switches Ethernet industrial, las estadísticas y el historial pueden ser usadas para identificar las tendencias de capacidad, otorgando a los usuarios la capacidad para identificar rápidamente los problemas y ver quiénes son los usuarios que utilizan más el ancho de banda de un solo vistazo. Utilización de unicast, multicast, broadcast y errores pueden ser graficados hasta 18 horas para los análisis a largo plazo.

1.6. CONTROLADORES DE AUTOMATIZACIÓN PROGRAMABLE

Los controladores de automatización programable es una tecnología orientada a la industria para el control automatizado, medición, diseño de controladores y prototipos, un PAC está conformado por un CPU, módulos de entradas, salidas y buses de datos ya sea uno o varios que a su vez interconectan entre todos.

Los PACs de National Instruments están diseñados con la tecnología de procesamiento más reciente para resolver problemas avanzados de control en entornos industriales y robustos. Un PAC de NI es completamente programable con NI LabVIEW, un entorno integrado de desarrollo gráfico.

NI CompactRIO es un PAC avanzado de National Instruments con

más de 60 módulos de E/S, más de 15 chasis y múltiples opciones de protocolos de comunicación [14].

- **Ventajas de los PACs de NI**

Entradas y Salidas de Gran Calidad a su Sistema de Control.

NI ha estado en la industria de medidas por más de 30 años y esa experiencia está aplicada en cada módulo de E/S de PAC. Los módulos analógicos están disponibles desde 12 hasta 24 bits y tienen velocidades de muestreo de DC hasta 1 MS/s/canal.

Los NI PACs son excelentes en sistemas de control analógico intensivos.

LabVIEW, el entorno de desarrollo gráfico para PACs de NI, maneja de una manera natural técnicas de programación más avanzadas, basadas en computación multinúcleo y de hilos múltiples. Esto le ayuda a programar sistemas de control que no serían posibles con una sintaxis más restringida como 61131.

Operación en Entornos Robustos (-40 °C a 70 °C, 5gRMS, 50g de impacto)

Los componentes CompactRIO son fabricados usando aluminio, no plástico y el chasis es un componente sólido colado a diferencia de los sistemas estilo expansión que tienen múltiples uniones.

El hardware de NI está diseñado para integrarse con sistemas existentes de PLCs, así usted puede usar CompactRIO para cubrir sus necesidades más complejas de control y medidas, y continuar usando PLCs estándares implementados actualmente.

1.6.1. NI CompactRIO

CompactRIO es un sistema embebido y reconfigurable de control y adquisición. La arquitectura robusta del hardware del sistema CompactRIO incluye módulos de E/S, un chasis FPGA reconfigurable y un controlador embebido. Además, CompactRIO se programa con herramientas de programación gráfica de NI LabVIEW y puede usarse en una variedad de aplicaciones de control y monitoreo embebidos.

Los sistemas CompactRIO consisten en un controlador embebido para comunicación y procesamiento, un chasis reconfigurable que aloja al FPGA programable por el usuario, módulos de E/S intercambiables en vivo y software gráfico LabVIEW para programación rápida en Windows, FPGA y sistemas en tiempo real mostradas en la Figura 1. 15.[14].



Figura 1. 15. NI CompactRIO. [14]

1.6.2. NI myRIO

El NI myRIO mostrado en la Figura 1. 16, pone en manos del usuario el procesador ARM® Cortex™-A9 dual-core en tiempo real y E/S personalizadas de FPGA Xilinx. Por medio de sus componentes internos, acceso a software transparente y biblioteca de recursos y tutoriales, NI myRIO es una herramienta accesible para el usuario, ideal para hacer una ingeniería real. [15]



Figura 1. 16. NI myRIO. [15]

1.6.2.1. Las Aplicaciones de NI myRIO:

NI myRIO es usado en una variedad de aplicaciones académicas, para mejorar el aprendizaje de la educación de ingeniería. Ya sea que se utilice solo o combinado con complementos, NI miniSystems o sensores de terceros, NI myRIO puede ayudar al aprendizaje de múltiples conceptos de ingeniería en un dispositivo.

Puede encontrar recursos, sobre las maneras en que NI myRIO es usado para la enseñanza en cada una de las siguientes áreas.

- Mecatrónica
- Control
- Robótica
- Embebido
- Desarrollo de Proyectos Avanzados

1.6.3. NI Single-Board RIO

La plataforma NI Single-Board RIO mostrada en la Figura 1. 17, combina dispositivos embebidos de despliegue el cual tiene un procesador en tiempo real, arreglo de compuertas programable en campo (FPGA) y E/S analógica y digital en una sola tarjeta programada con el software NI LabVIEW. Se

puede incrementar la E/S analógica y digital integrada usando módulos de la Serie C. Los dispositivos NI Single-Board RIO están diseñados para aplicaciones OEM embebidas de control y adquisición y de alto volumen que requieren alto rendimiento y fiabilidad.[16]

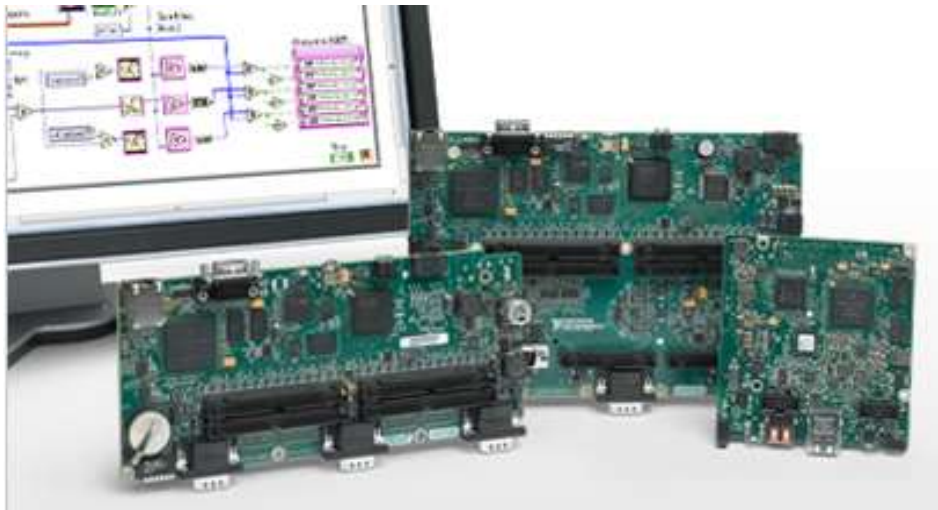


Figura 1. 17. NI Single-Board RIO. [16]

Requisitos de software

Se necesita un equipo de desarrollo con el siguiente software instalado en ella.

- LabVIEW
- Módulo LabVIEW Real-Time
- Módulo LabVIEW FPGA
- NI-RIO.

Requisitos de hardware

Se necesita del siguiente hardware para utilizar el dispositivo NI sbRIO.

- NI sbRIO-61x/9612XT/963x/9632XT/964x/9642XT
- 19-30 VCC
- Cable Ethernet

Aplicaciones

Son usando para simplificar el desarrollo, incremento de productividad y reducción considerable del tiempo al mercado.

Áreas de Aplicación de NI Single-Board RIO

- Control
- Energía/Potencia
- Robótica
- Monitoreo

1.6.3.1. Evaluación NI LabVIEW RIO

El Juego de Evaluación NI LabVIEW RIO mostrado en la Figura 1. 18, incluye lo necesario para experimentar el enfoque de desarrollo gráfico de sistemas de NI para sistemas embebidos. Este enfoque combina el software de diseño de sistemas LabVIEW y la plataforma estándar de hardware de E/S reconfigurable (RIO) basada en FPGA para reducir el tiempo en aplicaciones de control y monitoreo embebidos. [17].



Figura 1. 18. NI LabVIEW RIO Evaluation KIT. [17]

El Juego de Evaluación NI LabVIEW RIO está diseñado para el apoyo de evaluación de la plataforma Incluido hardware embebido objetivo con procesador RTOS, FPGA reconfigurable, E/S analógica y digital en la Tabla 1.1 se puede visualizar todas las características del KIT.

Tabla1. 1. Características del kit de evaluación NI LabVIEW RIO.

Características	Información adicional
Generador de funciones(cuadrada sinusoidal y triangular)	0 a 1 kHz, 0 a 10 kHz, o 0 a 100 kHz rangos de salida de frecuencia
Hitachi HD44780 LCD características del Display	Programado utilizando un controlador LabVIEW libre
Codificador de cuadratura	24 ciclos/rev
Potenciómetro	100K lineal
6 LED's programables por el usuario	
Sensor de temperatura	
6 entradas analógicas y 2 salidas analógicas	
5 pulsadores	
4 Líneas de E / S digitales	

Fuente: [17]

1.7. SOFTWARE DE DESARROLLO DE SISTEMAS NI LABVIEW

El software de programación para el desarrollo del controlador en este caso para el control difuso aplicado a los procesos de Presión y Flujo del Laboratorio de Redes Industriales y Control de Procesos es el software LabVIEW.

El software LabVIEW es ideal para cualquier sistema de medidas y control y el corazón de la plataforma de diseño de NI. Al integrar todas las herramientas que los ingenieros y científicos necesitan para construir una amplia variedad de aplicaciones en mucho menos tiempo, NI LabVIEW es un entorno de desarrollo para resolver problemas,

productividad acelerada y constante innovación. [18].

1.7.1. Módulo LabVIEW Real-Time

El Módulo NI LabVIEW Real-Time mostrada en la Figura 1. 19, es una solución completa para crear sistemas embebidos autónomos y confiables con un enfoque de programación gráfica. Como un complemento del entorno desarrollo de LabVIEW, el módulo le ayuda a desarrollar y depurar aplicaciones gráficas que usted puede descargar y ejecutar en dispositivos de hardware embebido como NI CompactRIO, NI Single-Board RIO, PXI, sistemas de visión o PCs de terceros.[19].



Figura 1. 19. Módulo LabVIEW Real-Time. [19]

- **Razones Principales para Usar el Módulo LabVIEW Real-time**

Programar gráficamente en LabVIEW puede mejorar enormemente su eficiencia y con el Módulo LabVIEW Real-Time usted puede usar este mismo enfoque para crear sistemas autónomos que se ejecutan por periodos prolongados de tiempo.

- **Sistema Operativo en Tiempo Real para Temporización Precisa y Alta Fiabilidad.**

Aunque LabVIEW es usado comúnmente para desarrollar aplicaciones que se ejecutan en SOs de escritorio como Windows, estos SOs no están optimizados para ejecutar aplicaciones críticas con estrictos requerimientos de temporización. El Módulo LabVIEW Real-Time tiene software de sistema operativo de tiempo real (RTOS) que se ejecuta en dispositivos de hardware embebido y ciertas PCs de terceros.

- **Amplia Variedad de IP y Controladores de Hardware en Tiempo Real.**

Con el Módulo LabVIEW Real-Time, se puede aprovechar cientos de bibliotecas LabVIEW pre-escritas incluyendo control algoritmo derivado proporcional integral (PID) y Transformada Rápida de Fourier (FFT) – en sus sistemas autónomos. Los controladores de hardware en tiempo real también están disponibles con la mayoría de los módulos de E/S, así se puede adquirir datos de manera determinística usando las llamadas io. De API (por ejemplo, NI-DAQmx) como en LabVIEW para PC de escritorio.

1.7.2. Módulo LabVIEW FPGA

NI LabVIEW mostrado en la Figura 1. 20, y el Módulo LabVIEW FPGA proporcionan un ambiente de desarrollo gráfico para chips FPGA en objetivos de hardware de E/S (RIO) reconfigurables. Con el Módulo LabVIEW FPGA, usted puede desarrollar VIs FPGA en un servidor ejecutando Windows y LabVIEW compila e implementa el código de hardware. Puede crear VIs de FPGA embebidos que combinan acceso directo a E/S con la lógica de LabVIEW definida por el usuario para definir hardware personalizado para aplicaciones como protocolos de

comunicación digital, simulación de control en hardware y rápida generación de prototipos de control.

El Módulo LabVIEW FPGA contiene varias rutinas integradas para procesar señales, puede integrar código existente de lenguaje de descripción de hardware (HDL), así como IP de terceros incluyendo funciones del Xilinx CORE Generator.

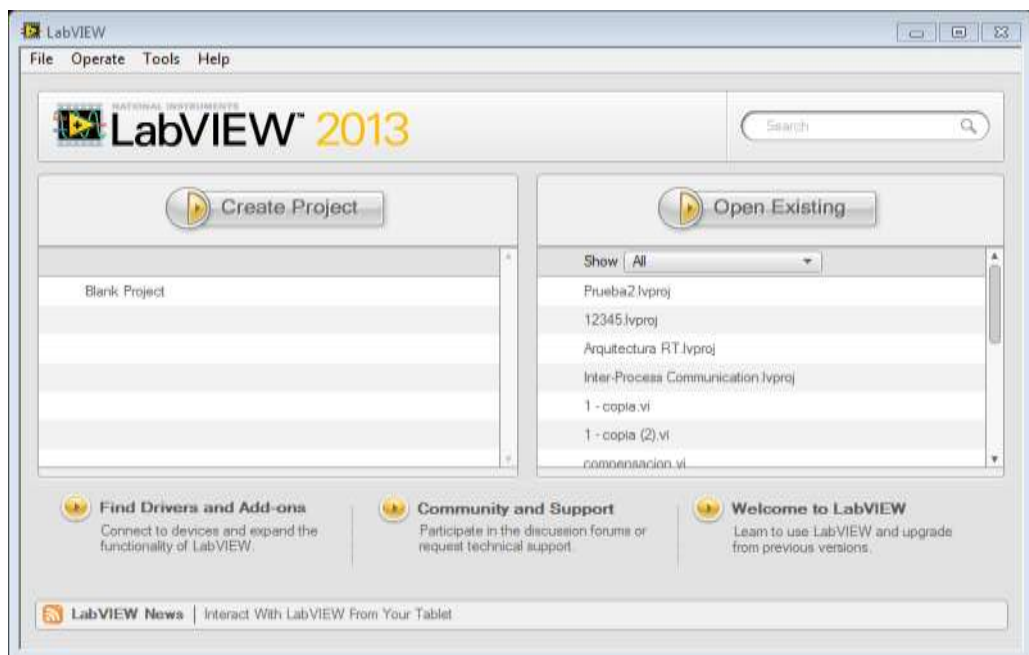


Figura 1. 20. NI LabVIEW

1.8. ESTACIONES DE PROCESOS

Las estaciones del laboratorio de Redes Industriales y Control de Procesos de la Universidad de las Fuerzas Armadas “ESPE” extensión Latacunga, fabricados por Lab-Volt, son estaciones de trabajo independientes implantadas para el aprendizaje práctico de las medidas, del control y reparaciones de **Presión, Flujo, temperatura y Nivel**.

Las estaciones pueden funcionar independientemente o también combinadas simulando procesos complejos existentes en la industria.
[20]

1.8.1. Estación De Flujo

La siguiente estación de Procesos de Flujo mostrada en la Figura 1. 21, representa el control del caudal de un líquido mediante la ayuda de una tubería. Para realizar el control de la estación de flujo se cuenta con un variador de frecuencia la cual maneja una bomba; esta toma la señal de 4 a 20 mA que es proveniente del controlador.

La bomba puede aumentar o disminuir el flujo de agua en el sensor tipo placa orificio según la orden deseada. En la tubería del proceso se cuenta con un transmisor de presión diferencial el cual se encarga de medir el flujo de agua y envía la señal medida al controlador con un rango de 4 a 20 mA. La misma que es una señal estandarizada.



Figura 1. 21. Estación de Proceso de Flujo. [20]

1.8.2. Estación De Presión

La siguiente estación de Procesos de Presión mostrada en la Figura 1. 22, consta de un proceso que incluye dos tanques de aire, la capacidad de cada uno es de dos galones, y se pueden conectar con cordones individualmente, en serie o en paralelo. Cuando se utiliza individualmente cualquiera de los dos tanques, se consigue un proceso de una sola capacidad con un solo volumen y un retraso de respuesta mínima. Cuando se relaciona dos tanques en paralelo se consigue un proceso de una capacidad, con volumen doble; cuando se relacionan los dos tanques en serie, el proceso tiene capacidad doble; ambos tienen un retraso de respuesta con efecto recíproco.

Esta estación también dispone con válvulas de aguja en las conexiones del tanque y en la carga, que permiten introducir perturbaciones en la demanda y el suministro de presión también permite variar el grado de restricciones en todas las configuraciones de circuitos cerrados, aportando velocidades y respuestas de proceso diversas.



Figura 1. 22. Estación de Proceso de Presión. [20]

1.8.3. Toolbox Sistema de Identificación de MATLAB

El toolbox contiene una aplicación para la construcción de modelos matemáticos de sistemas dinámicos a partir de los datos de entrada-salida medidos. Permite crear y utilizar modelos de sistemas dinámicos no fácilmente modelados a partir de primeros principios o especificaciones. Cuenta con la gran ventaja de poder utilizar los datos de entrada-salida en el dominio de tiempo y en dominio de la frecuencia para la identificación de las funciones de transferencia en tiempo discreto, tiempo continuo, modelos de procesos y modelos de espacio de estado. La herramienta de sistema de identificación mostrada en la Figura 1. 23, también proporciona algoritmos para la estimación de parámetros en línea incorporadas. [22]

Ofrece las técnicas de identificación, como máxima verosimilitud, la reducción al mínimo de predicción de errores (PEM) , y la identificación del sistema subespacial. Para representar la dinámica de sistemas no lineales, se puede estimar modelos Hammerstein-Wiener y modelos no

lineales ARX con la red wavelet, árbol-partición, y no linealidades red sigmoide. El toolbox realiza la identificación del sistema de color gris-box para la estimación de parámetros de un modelo definido por el usuario. Se puede utilizar el modelo identificado para la predicción de la respuesta del sistema y modelado de plantas en Simulink.

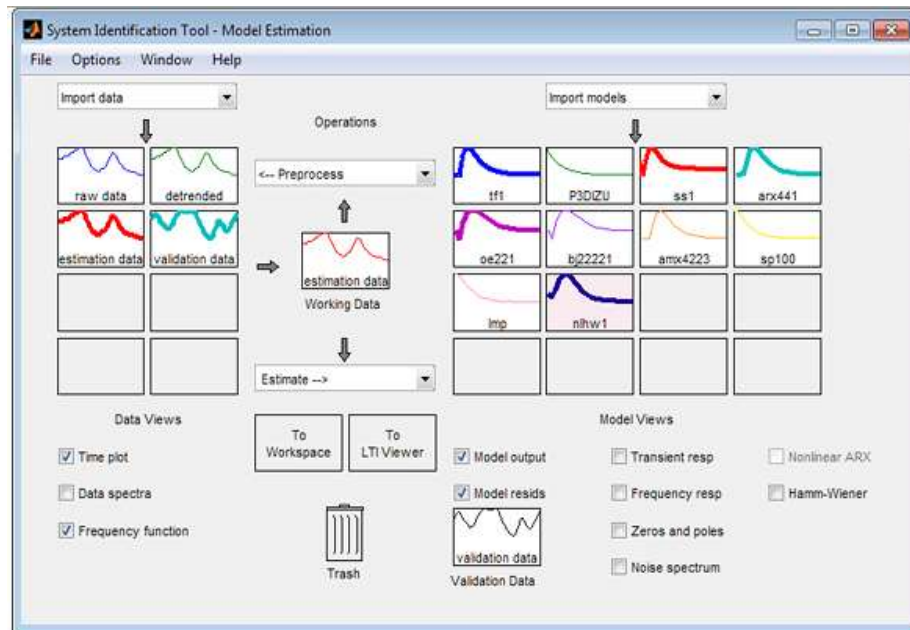


Figura 1. 23. Sistema de identificación. [22]

1.8.3.1. Identificación de modelo de datos en matlab.

Identificación System Toolbox permite crear modelos a partir de datos de entrada y salida medidos. Se puede:

- Determinar la estructura adecuada de modelo y de orden, y el modelo de estimación de parámetros.
- Analizar y procesar datos.
- Validar la exactitud del modelo.

- **Análisis y Procesamiento de Datos**

Para la identificación de modelos, es preciso especificar la información como el nombre de los canales de entrada y salida, tiempo de muestreo, y el comportamiento entre muestras, las cuales son ingresadas en la ventana mostrada en la Figura 1. 24, para la importación de datos. El toolbox permite conectar esta información con los datos, lo que facilita la visualización de la misma, conversión de dominio, y diversas tareas de preprocesamiento.

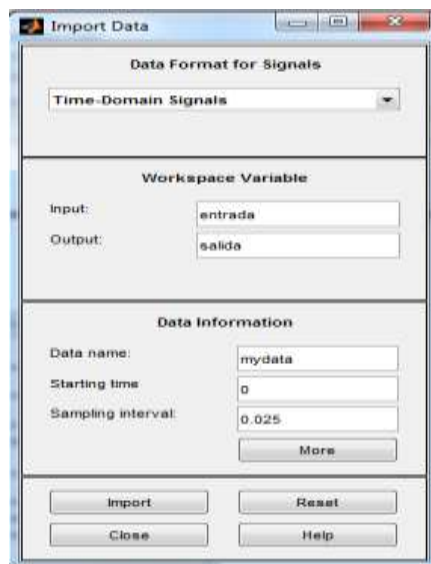


Figura 1. 24. Ventana para importar datos al ident. [22]

Los datos medidos a menudo tienen compensaciones, derivas lentas, valores atípicos, valores perdidos, y otras anomalías. El toolbox elimina este tipo de anomalías mediante operaciones tales como la realización detrending, filtrado, muestreo, y la reconstrucción de los datos que faltan operaciones realizadas en la etapa de preprocesamiento mostrada en la Figura 1. 25, el toolbox puede analizar la idoneidad de los datos para la identificación y proporcionar diagnósticos sobre la persistencia de la excitación, la existencia de ciclos de retroalimentación, y la presencia de no linealidades.

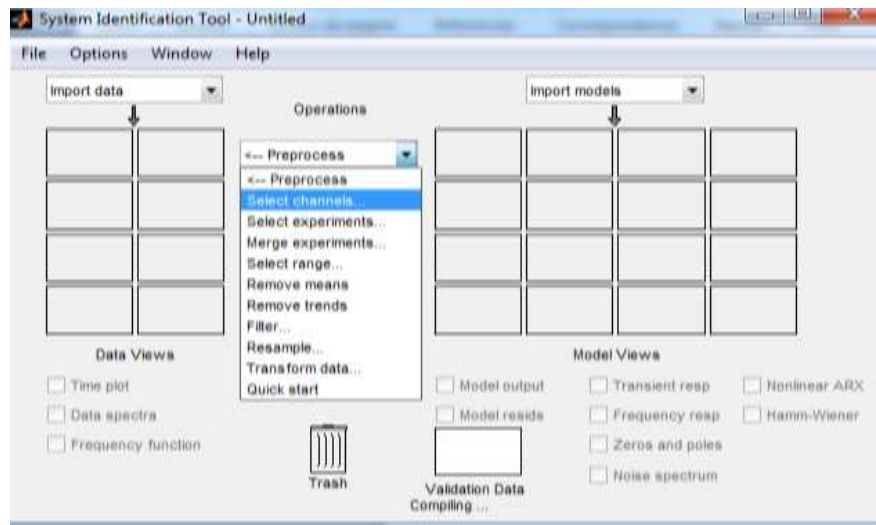


Figura 1. 25. Preprocesamiento de señal. [22]

El toolbox estima los impulsos y respuestas de frecuencia del sistema directamente a partir de datos medidos. El uso de estas respuestas, puede analizar las características del sistema, tales como constantes de tiempo dominante, retrasos de entrada, y las frecuencias de resonancia. También se puede utilizar estas características para configurar los modelos paramétricos durante la estimación.

- **Estimación de los parámetros del modelo**

Modelos paramétricos, tales como funciones de transferencia o modelos de espacio de estado mostradas en la Figura 1. 26, utilizan un pequeño número de parámetros para capturar la dinámica del sistema. System identification Toolbox estima los parámetros del modelo y sus incertidumbres de los datos de respuesta en frecuencia y de respuesta de tiempo. Puede analizar estos modelos utilizando respuesta en el tiempo y respuesta en frecuencia, como el paso, el impulso, diagramas de Bode, y los mapas de polos ceros.

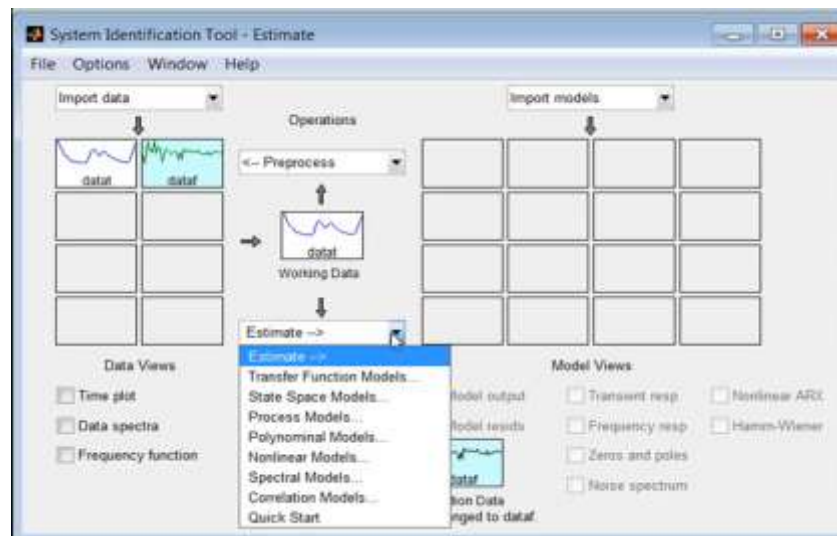


Figura 1. 26. Estimación y Validación de Modelos. [22]

- **Validación de Resultados**

El Toolbox ayuda a validar la precisión de los modelos identificados utilizando conjuntos independientes de datos medidos de un sistema real. Para un determinado conjunto de datos de entrada, el toolbox calcula la salida del modelo identificado y permite comparar esa salida con la salida medida a partir de un sistema real. También es posible ver el error de predicción y producir el tiempo de respuesta y la respuesta en frecuencia con límites de confianza para visualizar el efecto de las incertidumbres de los parámetros en las respuestas de los modelos.

CAPÍTULO 2

DISEÑO SIMULACIÓN E IMPLEMENTACIÓN DE LOS ALGORITMOS DE CONTROL

2.1. DISEÑO DEL CONTROLADOR PD DIFUSO MÁS ACCIÓN INTEGRAL DE LA PLANTA DE PRESIÓN.

Descripción de la Planta De Presión

La planta de presión consta de dos tanques, la capacidad de cada uno es de dos galones; cuando se relacionan los dos galones ya sea en serie o paralelo, se consigue un proceso con una capacidad de volumen doble.

La planta dispone de válvulas de aguja tanto en las conexiones del tanque como en la carga, las mismas que permiten provocar perturbaciones mediante un accionamiento manual.

El transmisor de la planta genera señales estandarizadas de corriente en el rango de 4-20 mA, permitiendo enviar las mismas al controlador usado. El actuador de la planta es una válvula de presión, que recibe señales estandarizadas de corriente (4-20mA), para posteriormente ser transformada en una señal neumática que abre o cierra la válvula.

Diseño del Algoritmo de Control Difuso para la planta de Presión

Para el diseño del controlador difuso se tomó en cuenta el siguiente procedimiento:

1.- Como primer punto se selecciona el tipo de controlador difuso teniendo como opciones el método de Mamdani y el de Takagi-Sugeno-Kangla, la estrategia escogida en este proyecto es el método de Mandami

asociada con principios PD (Proporcional, derivativo) más una acción integral a la salida del controlador difuso.

2.- Selección de las variables de entrada y salida del sistema. El sistema es de tipo MISO (Múltiples entradas, una salida), el mismo que cuenta con dos entradas y una salida; como entradas se tomó al error y a la derivada del error; en la Figura 2. 1, se muestra el diagrama de bloques del sistema.



Figura 2. 1. Diagrama de bloque del Control Difuso de la planta de presión

Elaborado por: Alomoto, L. y Quimbita, E.

3.- El tercer paso consiste en la definición de los Conjuntos difusos y las funciones de pertenencia de los espacios de entrada y salida. Para este caso se utiliza los conjuntos difusos de tipo triangulares y trapezoidales por ser las más comunes y más sencillas de realizar sus cálculos en comparación a los otros tipos, dentro de los mismos existirá el respectivo solapamiento como la teoría de Mandami sugiere.

Funciones de Pertenencia de las entradas

Para establecer los puntos de las funciones de pertenencia se toma en cuenta los rangos de medición de las entradas, en este caso el transmisor de corriente de la planta de presión envía señales que proporcionan al variar en el rango de 20 a 50 PSI.

Es así que las funciones de pertenencia del “error” se las diseño como se muestra la Figura 2. 2.

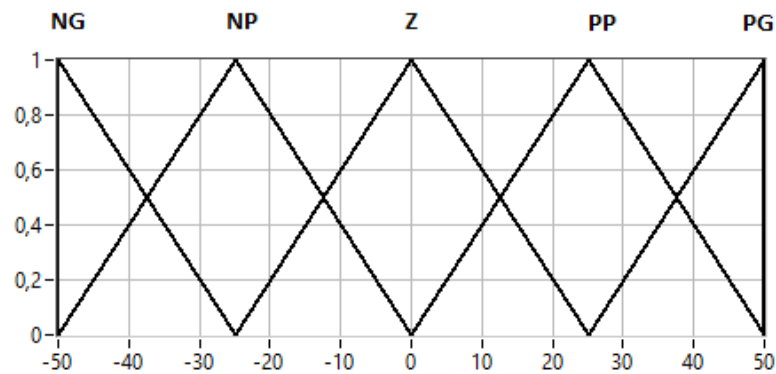


Figura 2. 2. Funciones de Pertenencia del error de la Planta de Presión

Elaborado por: Alomoto, L. y Quimbita, E.

Dónde:

NG, Negativo Grande

NP, Negativo Pequeño

Z, Cero

PP, Positivo Pequeño

PG, Positivo Grande

Las variables “error” y “derivada del error” usan similares funciones de pertenencia, ya que varían dentro del mismo rango; en la Figura 2. 3, se muestra las funciones de pertenencia para la derivada del error.

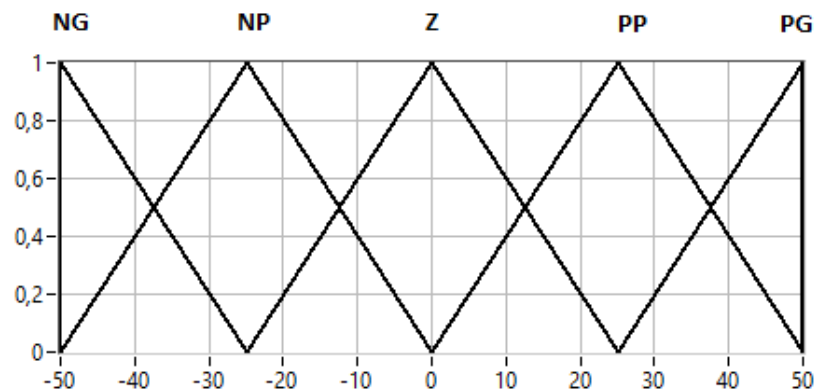


Figura 2. 3. Funciones de Pertenencia de la derivada del error de la Planta de Presión

Elaborado por: Alomoto, L. y Quimbita, E.

Dónde:

NG, Negativo Grande

NP, Negativo Pequeño

Z, Cero

PP, Positivo Pequeño

PG, Positivo Grande

Funciones de Pertenencia de la Salida

La salida del Sistema difuso o CV (Variable Controlada), incrementa o decrementa conforme la PV (Variable del Proceso) llega a un SP (Set Point). En el diseño se optó por aumentar el número de funciones de pertenencia en el conjunto difuso ya que gracias a la función NM (Negativo Medio) y PM (Positivo Medio) disminuyó considerablemente el error en estado estable. Por lo tanto a la salida, tomando en cuenta la rapidez por interacción del controlador, el rango de variación de la salida es de -1 a 1

Para la salida del sistema se utiliza funciones de pertenencia triangulares como se muestra en la Figura 2. 4.

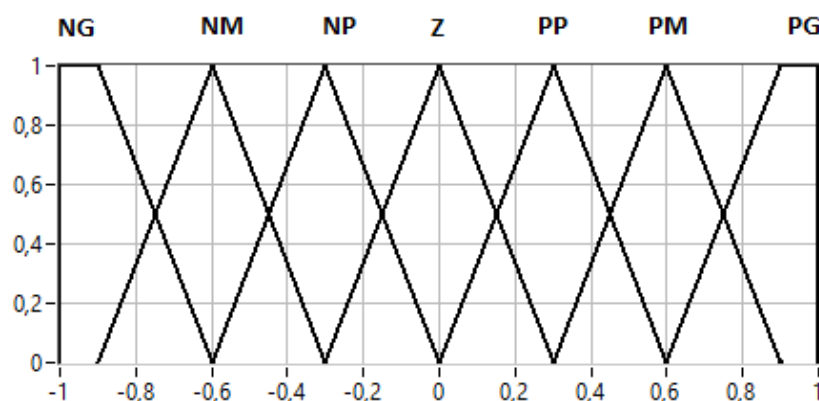


Figura 2. 4. Funciones de Pertenencia para la salida del sistema difuso de la planta de Presión.

Elaborado por: Alomoto, L. y Quimbita, E.

Dónde:

NG, Negativo Grande

NM, Negativo Medio

NS, Negativo Pequeño

Z, Cero

PS, Positivo Pequeño

PM, Positivo Medio

PB, Positivo Grande

4.- A continuación se define las reglas para el controlador difuso con respecto a las variables de entrada y salida, en este punto es muy importante usar la FAM (Matriz de reglas difusas).

Se partió de conceptos básicos como:

- Si el error es positivo la salida es positiva.
- Si el error es negativo la salida es negativa.
- Si el error es cero la salida es cero.

A partir de estos conceptos se relaciona el error con la, para tener una salida deseada.

En la Tabla 2. 1 se tiene la FAM para la planta de presión.

Tabla 2. 1. FAM del Controlador de planta de Presión.

de\e	NG	NP	Z	PP	PG
NG	NG	NG	NM	NP	Z
NB	NG	NM	NP	Z	PP
Z	NM	NP	Z	PP	PM
PP	NP	Z	PP	PM	PG
PG	Z	PP	PM	PG	PG

Elaborado por: Alomoto, L. y Quimbita, E.

5.- Para la obtención de la respuesta deseada se realiza un proceso inverso a la fuzzificación. Para realizar este proceso existen varios métodos, pero ninguno se considera más apropiado que otro, por este motivo el usuario puede elegir su método de acuerdo a los requerimientos que exige el sistema. Es así que en este sistema se eligió el método del Centro de gravedad por la razón de ser el más común, eficiente y el más utilizado en controles PID Difusos.

2.2. DISEÑO DEL CONTROLADOR PID DIFUSO MÁS ACCIÓN INTEGRAL DE LA PLANTA DE FLUJO.

Descripción de la Planta De Flujo

La planta cuenta con un transmisor de presión diferencial que envía una señal estándar de corriente de 4-20mA, esta señal es captada por un controlador, esta señal puede ser transformada a una señal de voltaje estándar de 1-5v, gracias a la resistencia de 250 ohmios que posee la propia planta.

Esta planta usa como actuador un variador de frecuencia, el mismo que realiza la función de control de la bomba de la planta; el actuador es accionado gracias a la de señal de corriente de 4-20mA que esta recibe.

Diseño del Algoritmo de Control Difuso para la planta de Flujo

Para el diseño del controlador difuso de la planta de flujo se tomó en cuenta el siguiente procedimiento:

1.- Al igual que el diseño de la planta de Presión como primer punto la estrategia de fuzzificación seleccionada para este algoritmo es la de Mandami con la diferencia que esta asocia principios PID (Proporcional, derivativo, integral) más una acción integral a la salida del controlador difuso.

2.- Selección de las variables de entrada y salida del sistema. El sistema es de tipo MISO, este cuenta con tres entradas y una salida. Como entradas se tomó al “error”, “derivada del error”, y a la “integral del error”, en la Figura 2. 5, podemos observar cómo se distribuyen las entradas y las salidas en nuestro sistema difuso.



Figura 2. 5. Diagrama de bloque del Control Difuso de la planta de presión

Elaborado por: Alomoto, L. y Quimbita, E.

3.- El tercer paso consiste en la definición de los Conjuntos difusos y las funciones de pertenencia de los espacios de entrada y salida. Para este caso se utilizó los conjuntos difusos de tipo triangulares con su respectivo solapamiento.

Funciones de Pertenencia de las entradas

El rango de medición para la planta de flujo varía entre 10 a 45 inH₂O, por lo tanto las funciones de pertenencia del error se las diseño como se muestra la Figura 2. 6.

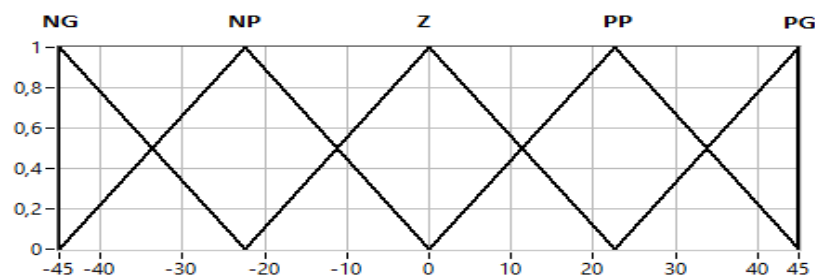


Figura 2. 6. Funciones de Pertenencia del error de la Planta de Flujo

Elaborado por: Alomoto, L. y Quimbita, E.

Dónde:

NG, Negativo Grande

NP, Negativo Pequeño

Z, Cero

PP, Positivo Pequeño

PG, Positivo Grande

Las variables “error”, “derivada del error”, e “integral del error” usan similares funciones de pertenencia, esto debido a que varían dentro del mismo rango.

En la Figura 2. 7, se muestra las funciones de pertenencia para la derivada del error de la planta de flujo.

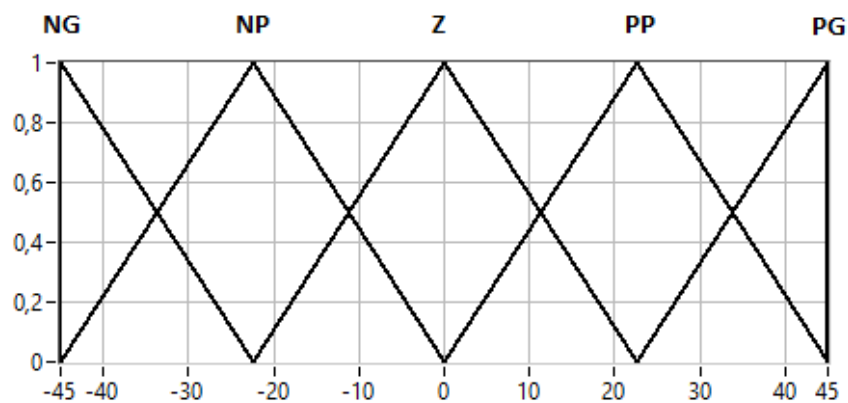


Figura 2. 7. Funciones de Pertenencia de la Derivada del error de la Planta de Flujo

Elaborado por: Alomoto, L. y Quimbita, E.

Dónde:

NG, Negativo Grande

NP, Negativo Pequeño

Z, Cero

PP, Positivo Pequeño

PG, Positivo Grande

En la Figura 2. 8, se muestra las funciones de pertenencia para la integral del error de la planta de flujo.

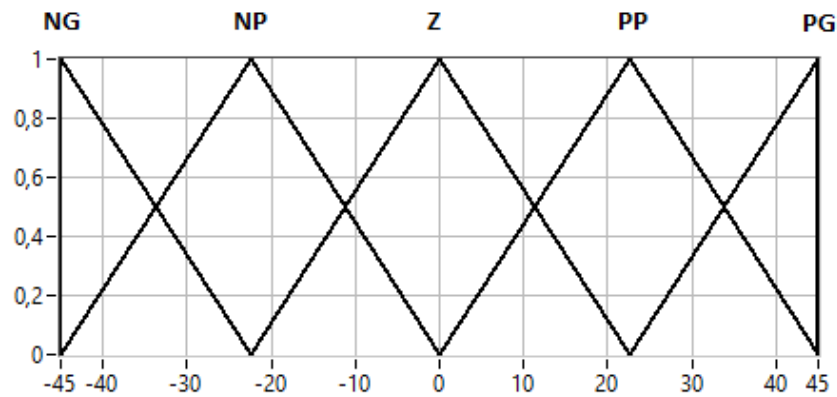


Figura 2. 8. Funciones de Pertenencia de la Integral del error de la Planta de Flujo

Elaborado por: Alomoto, L. y Quimbita, E.

Dónde:

NG, Negativo Grande

NP, Negativo Pequeño

Z, Cero

PP, Positivo Pequeño

PG, Positivo Grande

Funciones de Pertenencia de las Salidas

La variable salida del Sistema de flujo varia en el número de funciones de pertenencia en el conjunto difuso, la salida del Sistema CV, incrementa o decrementa conforme la PV llega al SP. Por lo tanto a la salida, tomando en cuenta la rapidez por interacción del controlador, el rango de variación de la salida es de -0,5 a 0,5.

Para la salida del sistema se utiliza funciones de pertenencia triangulares como se muestra en la Figura 2. 9.

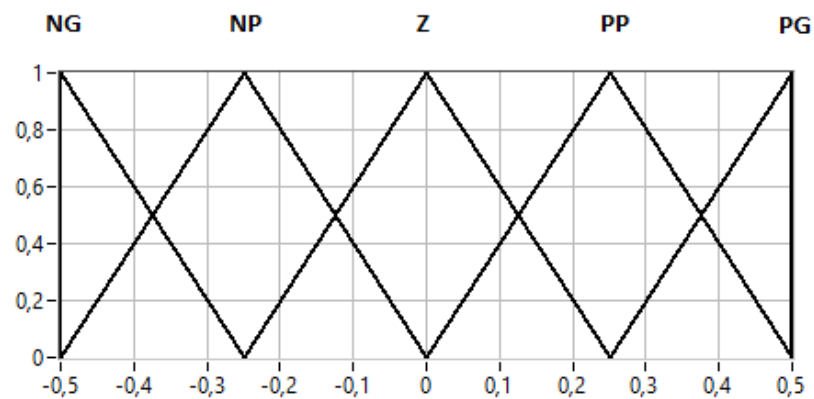


Figura 2. 9. Función de Pertenenca para la salida del sistema difuso para la planta de Flujo.

Elaborado por: Alomoto, L. y Quimbita, E.

Dónde:

NG, Negativo Grande

NP, Negativo Pequeño

Z, Cero

PP, Positivo Pequeño

PG, Positivo Grande

4.- La FAM para la planta de flujo consta de 125 reglas que relacionan el error la derivada del error y la integral del error, la Tabla 2. 2 muestra la relación entre las entradas y salidas para el proceso de flujo.

Tabla 2. 2. FAM del Controlador de planta de Flujo.

# Regla	Error	Derivada	Integral	Salida
1	NG	NG	NG	NG
2	NG	NG	NP	NG
3	NG	NG	ZE	NG
4	NG	NG	PP	NG
5	NG	NG	PG	NG
6	NG	NP	NG	NG
7	NG	NP	NP	NG

8	NG	NP	ZE	NG
9	NG	NP	PP	NG
10	NG	NP	PG	NG
11	NG	ZE	NG	NG
12	NG	ZE	NP	NG
13	NG	ZE	ZE	NG
14	NG	ZE	PP	NG
15	NG	ZE	PG	NG
16	NG	PP	NG	NP
17	NG	PP	NP	NP
18	NG	PP	ZE	NP
19	NG	PP	PP	NP
20	NG	PP	PG	NP
21	NG	PG	NG	NG
22	NG	PG	NP	NG
23	NG	PG	ZE	NG
24	NG	PG	PP	NG
25	NG	PG	PG	NG
26	NP	NG	NG	NP
27	NP	NG	NP	NP
28	NP	NG	ZE	NP
29	NP	NG	PP	NP
30	NP	NG	PG	NP
31	NP	NP	NG	NP
32	NP	NP	NP	NP
33	NP	NP	ZE	NP
34	NP	NP	PP	NP
35	NP	NP	PG	NP
36	NP	ZE	NG	NP
37	NP	ZE	NP	NP
38	NP	ZE	ZE	NP
39	NP	ZE	PP	NP

40	NP	ZE	PG	NP
41	NP	PP	NG	NP
42	NP	PP	NP	NP
43	NP	PP	ZE	NP
44	NP	PP	PP	NP
45	NP	PP	PG	NP
46	NP	PG	NG	NP
47	NP	PG	NP	NP
48	NP	PG	ZE	NP
49	NP	PG	PP	NP
50	NP	PG	PG	NP
51	ZE	NG	NG	ZE
52	ZE	NG	NP	ZE
53	ZE	NG	ZE	ZE
54	ZE	NG	PP	ZE
55	ZE	NG	PG	ZE
56	ZE	NP	NG	ZE
57	ZE	NP	NP	ZE
58	ZE	NP	ZE	ZE
59	ZE	NP	PP	ZE
60	ZE	NP	PG	ZE
61	ZE	ZE	NG	ZE
62	ZE	ZE	NP	ZE
63	ZE	ZE	ZE	ZE
64	ZE	ZE	PP	ZE
65	ZE	ZE	PG	ZE
66	ZE	PP	NG	ZE
67	ZE	PP	NP	ZE
68	ZE	PP	ZE	ZE
69	ZE	PP	PP	ZE
70	ZE	PP	PG	ZE
71	ZE	PG	NG	ZE

72	ZE	PG	NP	ZE
73	ZE	PG	ZE	ZE
74	ZE	PG	PP	ZE
75	ZE	PG	PG	ZE
76	PP	NG	NG	PP
77	PP	NG	NP	PP
78	PP	NG	ZE	PP
79	PP	NG	PP	PP
80	PP	NG	PG	PP
81	PP	NP	NG	PP
82	PP	NP	NP	PP
83	PP	NP	ZE	PP
84	PP	NP	PP	PP
85	PP	NP	PG	PP
86	PP	ZE	NG	PP
87	PP	ZE	NP	PP
88	PP	ZE	ZE	PP
89	PP	ZE	PP	PP
90	PP	ZE	PG	PP
91	PP	PP	NG	PP
92	PP	PP	NP	PP
93	PP	PP	ZE	PP
94	PP	PP	PP	PP
95	PP	PP	PG	PP
96	PP	PG	NG	PP
97	PP	PG	NP	PP
98	PP	PG	ZE	PP
99	PP	PG	PP	PP
100	PP	PG	PG	PP
101	PG	NG	NG	PG
102	PG	NG	NP	PG
103	PG	NG	ZE	PG

104	PG	NG	PP	PG
105	PG	NG	PG	PG
106	PG	NP	NG	PP
107	PG	NP	NP	PP
108	PG	NP	ZE	PP
109	PG	NP	PP	PP
110	PG	NP	PG	PP
111	PG	ZE	NG	PG
112	PG	ZE	NP	PG
113	PG	ZE	ZE	PG
114	PG	ZE	PP	PG
115	PG	ZE	PG	PG
116	PG	PP	NG	PG
117	PG	PP	NP	PG
118	PG	PP	ZE	PG
119	PG	PP	PP	PG
120	PG	PP	PG	PG
121	PG	PG	NG	PG
122	PG	PG	NP	PG
123	PG	PG	ZE	PG
124	PG	PG	PP	PG
125	PG	PG	PG	PG

Fuente: [12]

5.- Al igual que en la Planta de Presión el método de defuzzificación utilizado es el de Centro de Gravedad debido a su uso común para este tipo de controles.

2.3. SIMULACIÓN DE LOS ALGORITMOS DE CONTROL DIFUSOS DE LAS PLANTAS DE PRESIÓN Y FLUJO

Las herramientas usadas para la simulación son el Simulink y el Toolbox Fuzzy Logic de MATLAB. Esta herramienta es útil ya que permite simular los algoritmos de control, además permite realizar acciones experimentales con los modelos matemáticos obtenidos. Estas ventajas son aprovechadas para realizar las respectivas comparaciones entre un Control Difuso PID vs un Control PID Tradicional.

2.3.1. Simulación del Algoritmo de Control de Presión

Matlab cuenta con el Toolbox Fuzzy Logic que permite interactuar fácilmente con el usuario, como se muestra en la Figura 2. 10, el sistema es de tipo MISO (Múltiples entradas, y una salida)

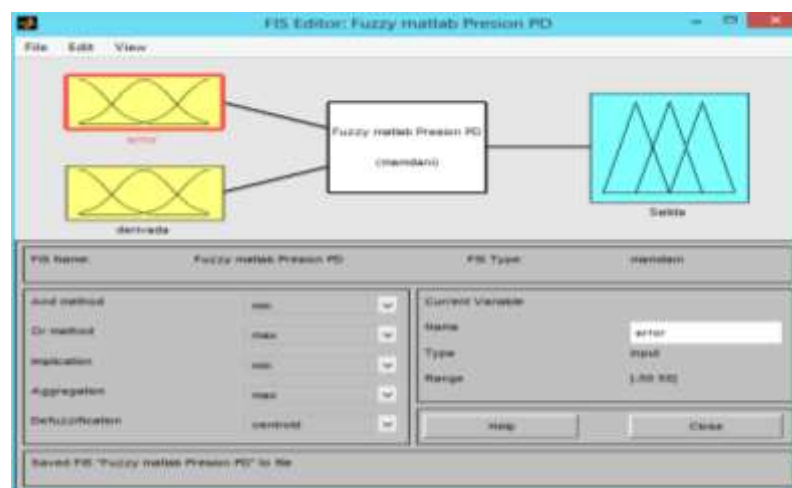


Figura 2. 10. Fuzzy en Matlab de la planta Presión

Elaborado por: Alomoto, L. y Quimbita, E.

La variable “error” consta de cinco funciones de pertenencia de tipo triangular, en la Figura 2. 11, se muestra como ingresar en el Toolbox el conjunto difuso “error” con sus respectivas funciones de pertenencia.

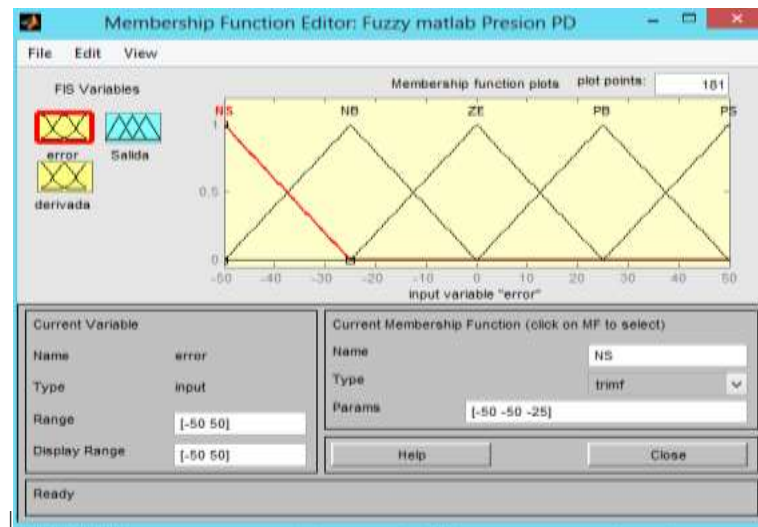


Figura 2. 11. Conjunto Difuso "Error" de la planta de Presión

Elaborado por: Alomoto, L. y Quimbita, E.

El conjunto difuso “derivada” que representa la derivada del error se muestra en la Figura 2. 12.

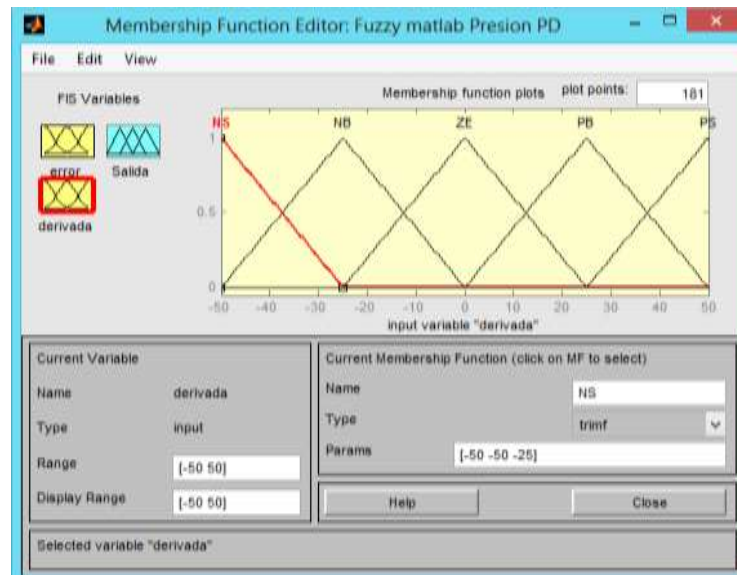


Figura 2. 12. Conjunto Difuso "derivada" de la planta de Presión

Elaborado por: Alomoto, L. y Quimbita, E.

El conjunto difuso "Salida" se muestra en la Figura 2. 13.



Figura 2. 13. Conjunto Difuso "Salida" de la planta de Presión

Elaborado por: Alomoto, L. y Quimbita, E.

Las reglas se ingresan de forma sencilla, la interface permite modificar y borrar reglas, además permite escoger el método de defuzzificación como se muestra en la Figura 2. 14.

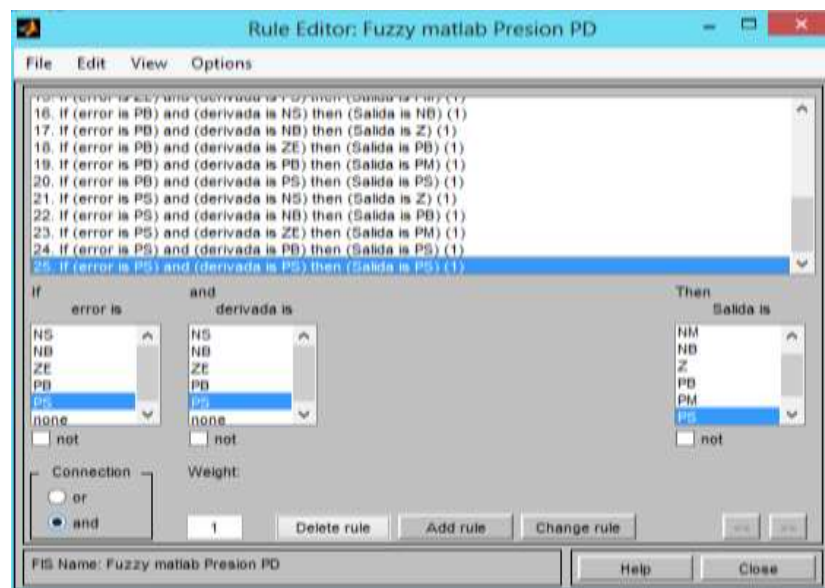


Figura 2. 14. Reglas del Controlador de la Planta de Presión.

Elaborado por: Alomoto, L. y Quimbita, E.

El toolbox permite visualizar como actúa el controlador antes de ser integrado al sistema de control, como se muestra en la Figura 2. 15.

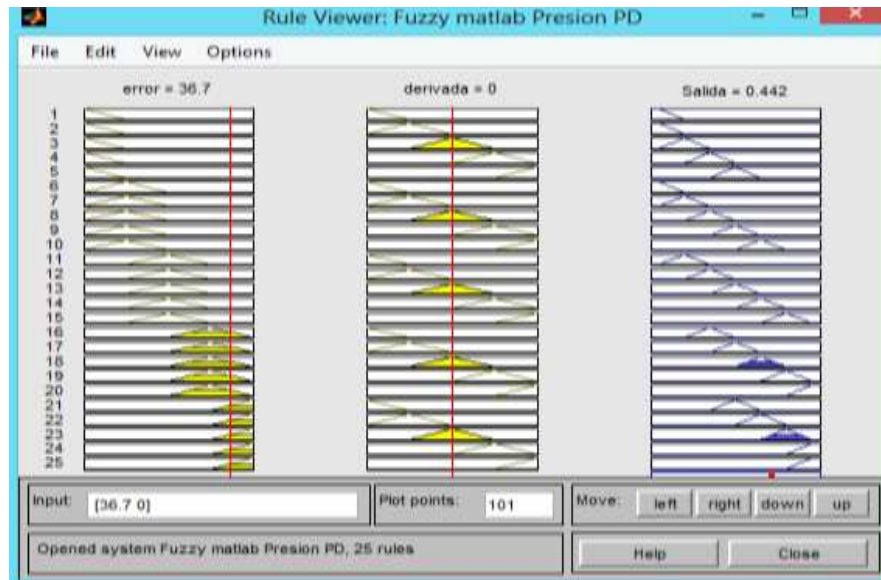


Figura 2. 15. Visor de Reglas de la Planta de Presión

Elaborado por: Alomoto, L. y Quimbita, E.

El Surface Viewer (Visualizador de Superficies) es un gráfico que muestra la superficie de la salida respecto a las entradas “error” y “derivada”; en la Figura 2. 16, se observa el Surface Viewer para la planta de Presión.

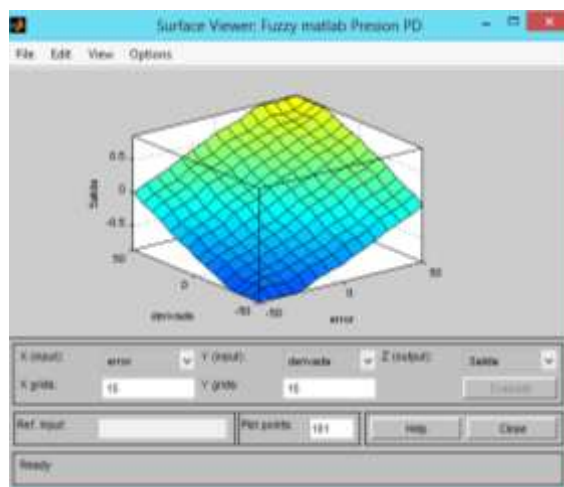


Figura 2. 16. Surface Viewer para la Planta de Presión

Elaborado por: Alomoto, L. y Quimbita, E.

2.3.2. Simulación del Algoritmo de Control de Flujo

El sistema para la planta de Flujo es un sistema MISO, teniendo como entradas al error, la derivada del error, y la integral del error, y una salida como se muestra en la Figura 2. 17.

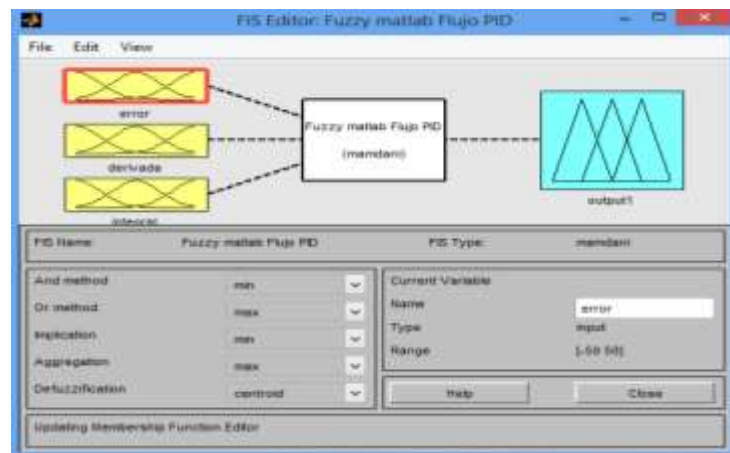


Figura 2. 17. Fuzzy en Matlab de la planta de Flujo

Elaborado por: Alomoto, L. y Quimbita, E.

La variable “error” consta de cinco funciones de pertenencia de tipo triangular, en la Figura 2. 18, se muestra el conjunto difuso “error” con sus respectivas funciones de pertenencia.

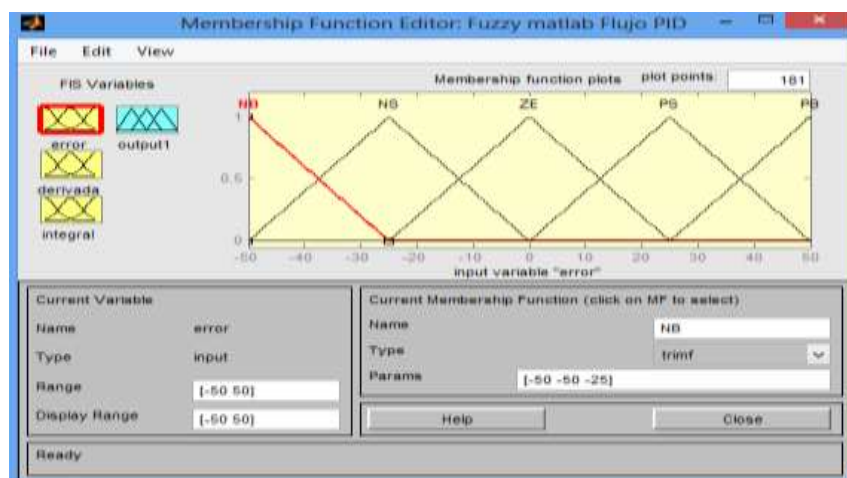


Figura 2. 18. Conjunto Difuso "Error" de la planta de Flujo

Elaborado por: Alomoto, L. y Quimbita, E.

El conjunto difuso “derivada” que representa la derivada del error se muestra en la Figura 2. 19.

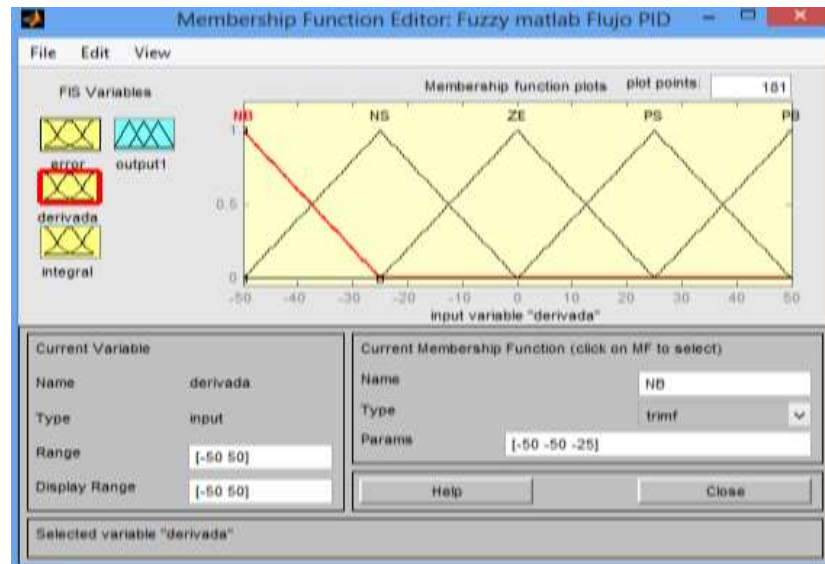


Figura 2. 19. Conjunto Difuso "derivada" de la planta de Flujo

Elaborado por: Alomoto, L. y Quimbita, E.

El conjunto difuso “integral” que representa la integral del error se muestra en la Figura 2. 20.

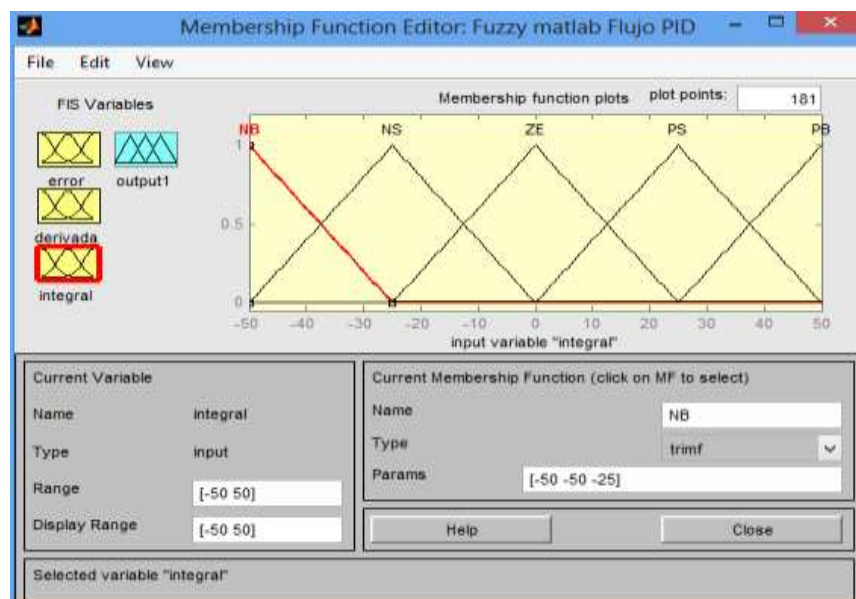


Figura 2. 20. Conjunto Difuso "integral" de la planta de Flujo

Elaborado por: Alomoto, L. y Quimbita, E.

El conjunto difuso "Salida" se muestra en la Figura 2. 21.

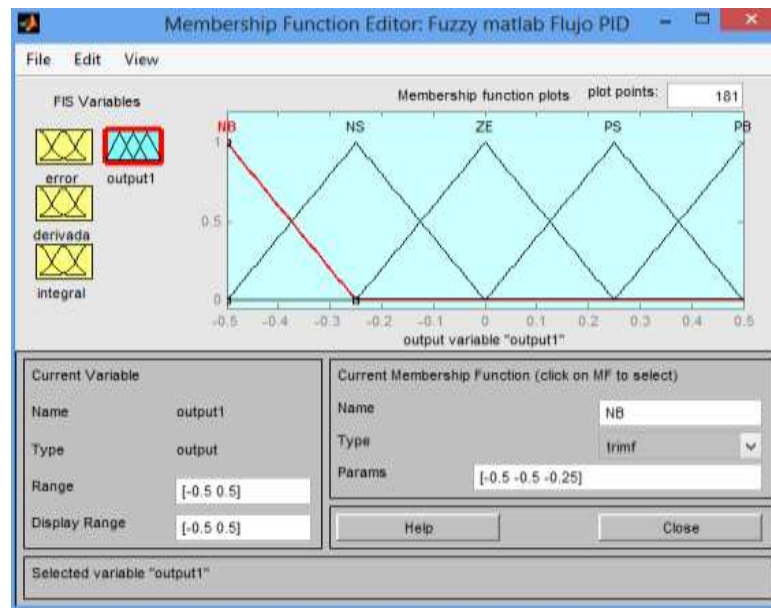


Figura 2. 21. Conjunto Difuso "Salida" de la planta de Flujo

Elaborado por: Alomoto, L. y Quimbita, E.

Las reglas se ingresan de forma sencilla, para la planta de flujo son 125 reglas que relacionan al error, la derivada del error, y a la integral del error, la interface permite modificar y borrar reglas, además permite escoger el método de defuzzificación como se muestra en la Figura 2. 22.

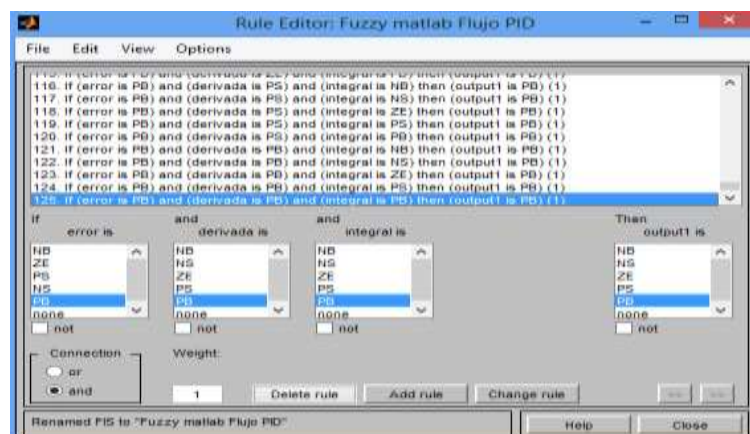


Figura 2. 22. Reglas Del Controlador de la Planta de Flujo.

Elaborado por: Alomoto, L. y Quimbita, E.

El toolbox permite visualizar como actúa el controlador antes de ser integrado al sistema de control, en la Figura 2. 23, se muestra cómo actúan las reglas anteriormente diseñadas.

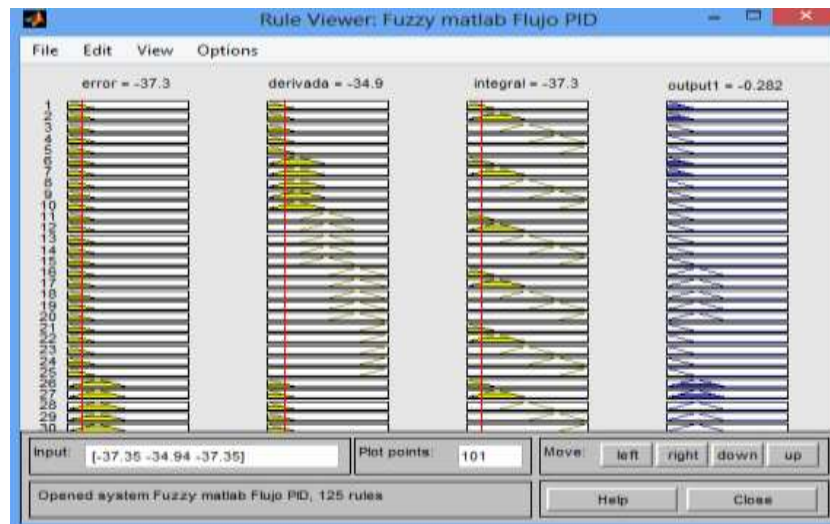


Figura 2. 23. Visor de Reglas de la Planta de Flujo

Elaborado por: Alomoto, L. y Quimbita, E.

El Surface Viewer para la planta de Flujo muestra la superficie de la salida respecto a las entradas “error”, “derivada”, e “integral”; en la Figura 2. 24, se observa el Visualizador de Superficie para la planta de Flujo.

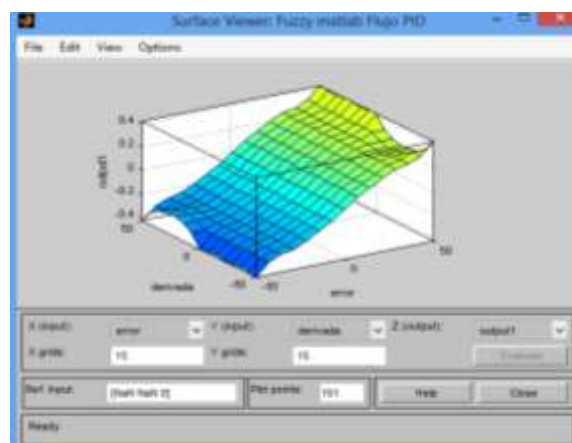


Figura 2. 24. Surface Viewer para la Planta de Flujo

Elaborado por: Alomoto, L. y Quimbita, E.

2.4. SIMULACIÓN A LAZO CERRADO DE LOS ALGORITMOS DE CONTROL DE PRESIÓN Y FLUJO.

Para comprobar el funcionamiento en lazo cerrado del controlador se usa el software de MATLAB específicamente la herramienta llamada Simulink, además de un modelo matemático de la planta que se obtiene gracias al método de identificación que se explica en el apartado [1.8.3].

A continuación se muestra la simulación de los controladores diseñados en lazo cerrado.

2.4.1. Planta de Presión

En la Figura 2. 25, se aprecia el controlador Fuzzy PD + acción Integral aplicada a la planta de presión, donde la acción integral permite tener una CV (Variable de Control) equilibrada, reduciendo las oscilaciones a la salida del controlador.

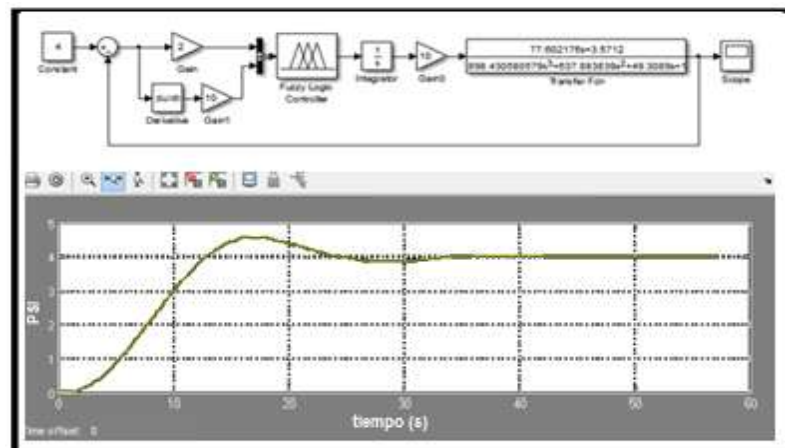


Figura 2. 25. Simulación del Algoritmo de Control Difuso de Presión

Elaborado por: Alomoto, L. y Quimbita, E.

2.4.2. Planta de Flujo

En la Figura 2. 26, se aprecia el controlador Fuzzy PID + acción Integral aplicada a la planta de flujo, donde la acción integral permite tener una CV (Variable de Control) equilibrada, reduciendo las oscilaciones a la salida del controlador.

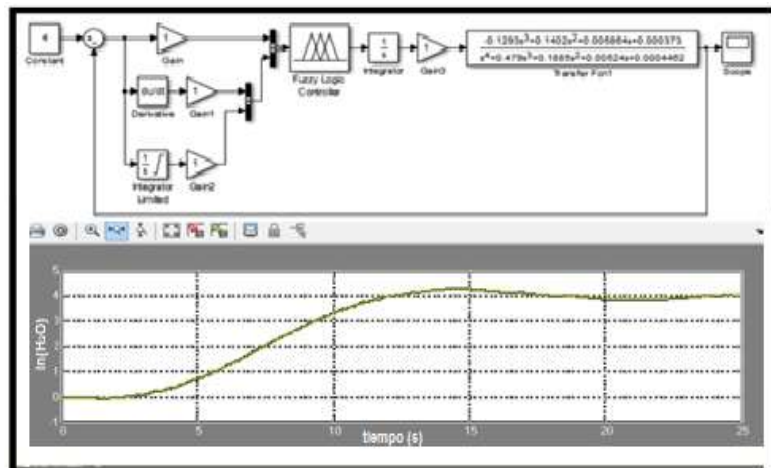


Figura 2. 26. Simulación del Algoritmo de Control Difuso de Flujo

Elaborado por: Alomoto, L. y Quimbita, E.

2.5. IMPLEMENTACIÓN DE LOS ALGORITMOS DE CONTROL DIFUSOS DE PRESIÓN Y FLUJO.

Creación de Proyecto en el dispositivo de control NI SB-RIO

El software Labview dispone de proyectos para tarjetas NI SB-RIO, por tal motivo es sencillo iniciar una aplicación en estas tarjetas. En la Figura 2. 27, se muestra como empezar un proyecto con NI SB-RIO.

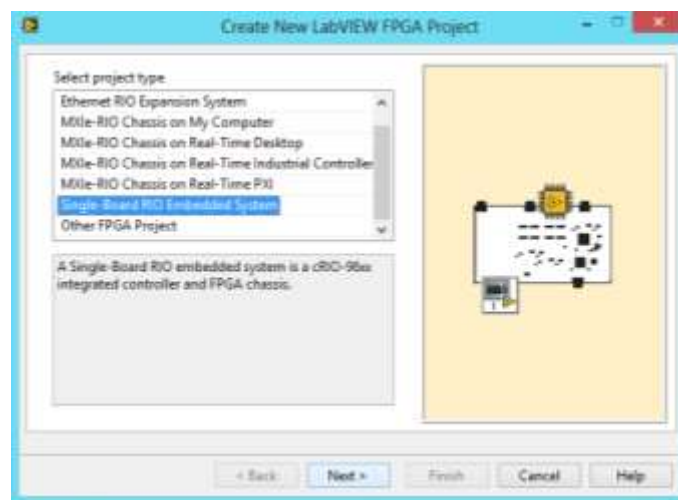


Figura 2. 27. Nuevo Proyecto FPGA's

Elaborado por: Alomoto, L. y Quimbita, E.

El controlador consta dos partes la FPGA y la RT, en este proyecto se utilizó la FPGA para la adquisición y la generación de entradas y salidas respectivamente, y la RT para la programación del Controlador Difuso.

Programación de la FPGA (Entradas y Salidas)

Para la programación de las entradas y salidas se utiliza el panel de selección, aquí se elige el tipo de entradas o salidas ya sean digitales o analógicas como podemos ver en la Figura 2. 28.

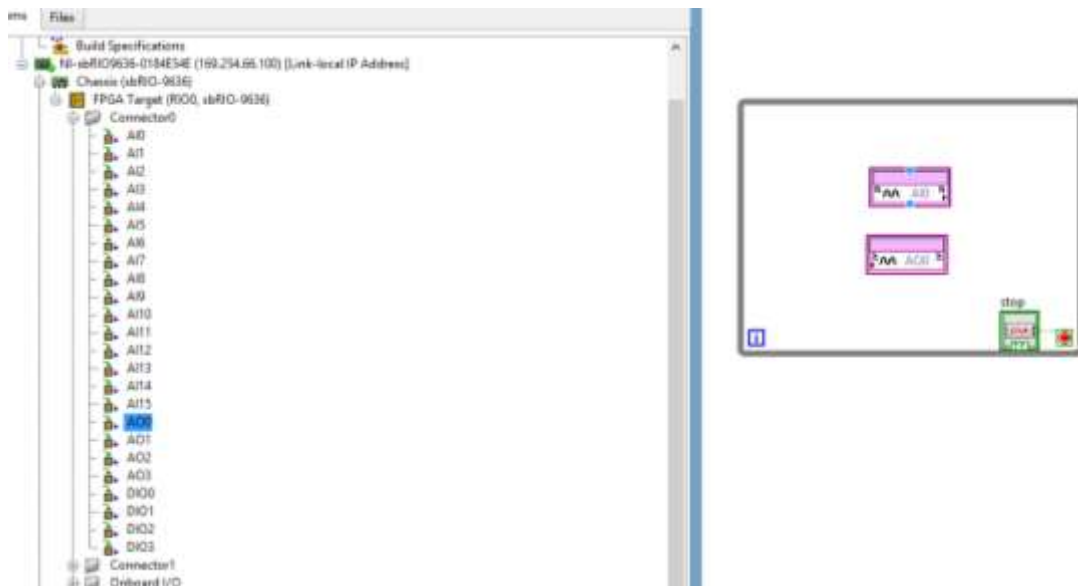


Figura 2. 28. Panel de selección de entradas/salidas digitales y analógicas

Elaborado por: Alomoto, L. y Quimbita, E.

Programación en el Real-Time (RT).

En el VI del RT, se direcciona el VI de entradas y salidas de la FPGA, luego de hacer un reset, inicia la programación del lazo WHILE, a continuación se toma datos para procesarlos según los requerimientos del usuario, finalmente se cierra la aplicación de la FPGA. En la Figura 2. 29. Se puede observar un ejemplo simple de programación con RT.

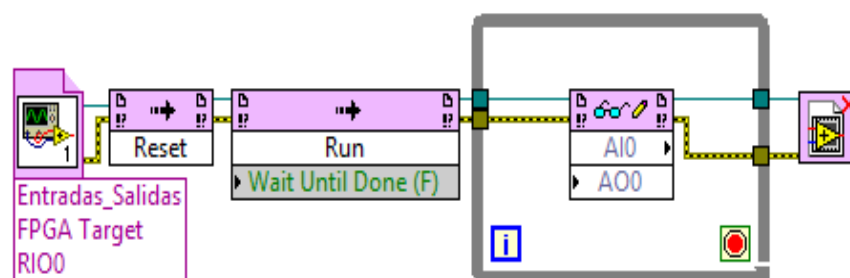


Figura 2. 29. Programación Real-Time (RT)

Elaborado por: Alomoto, L. y Quimbita, E.

El algoritmo de control es programado dentro del VI del RT. En este caso ya no se utiliza la herramienta Fuzzy System Designer debido a que el controlador es una FPGA, y dicha herramienta no funciona en estas tarjetas debido a que no se reconoce el tipo de archivo que se genera el Fuzzy System Designer; para solucionar este inconveniente el controlador difuso se programó únicamente con el toolbox Fuzzy Logic, como se muestra en la Figura 2. 30.

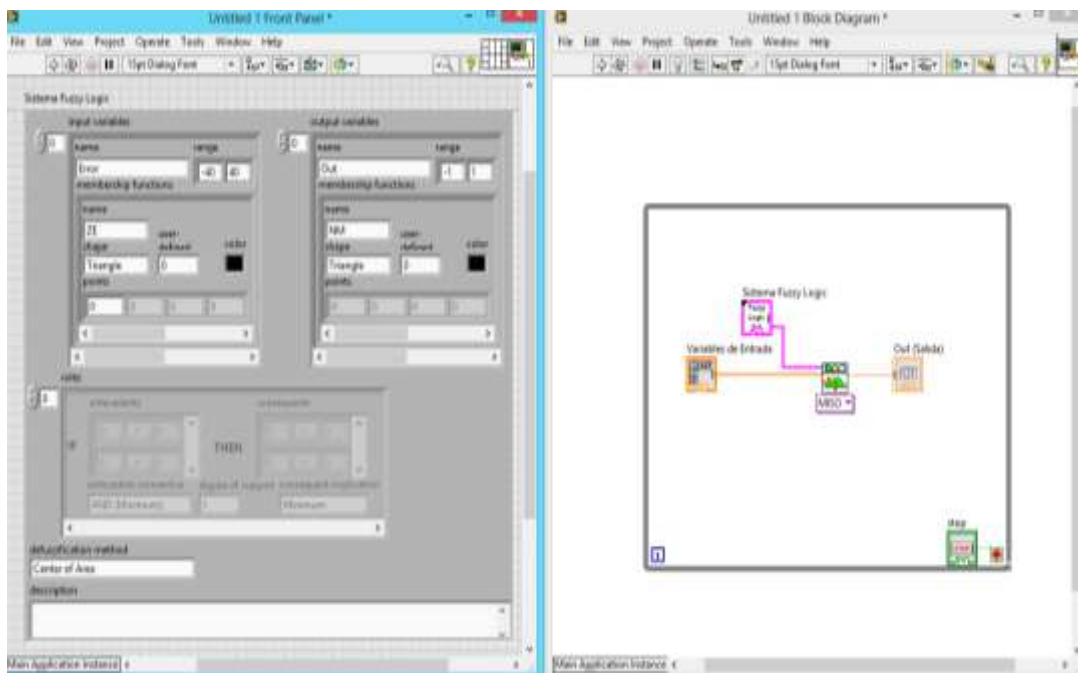


Figura 2. 30. Programación Control Difuso

Elaborado por: Alomoto, L. y Quimbita, E.

Integración del Conversor Voltaje-Corriente

En este proyecto se necesita un conversor de Voltaje/Corriente puesto que las plantas usadas tienen actuadores que reciben señales de corriente de entre 4-20 mA, y la tarjeta SB-RIO únicamente dispone de salidas análogas de voltaje. Por esta razón se añade a la salida de la FPGA el conversor que se lo puede visualizar en la Figura 2. 31.



Figura 2. 31. Conversor Voltaje-Corriente

Elaborado por: Alomoto, L. y Quimbita, E.

A continuación en la Figura 2. 32, se puede visualizar la implementación del algoritmo de control difuso de presión en la tarjeta NI SB-RIO siendo monitoreado por una PC.



Figura 2. 32. Implementación del Control Difuso en la Planta de Presión

Elaborado por: Alomoto, L. y Quimbita, E.

En la Figura 2. 33, se puede visualizar la implementación del algoritmo de control difuso de flujo en la tarjeta NI SB-RIO siendo monitoreado por una PC.

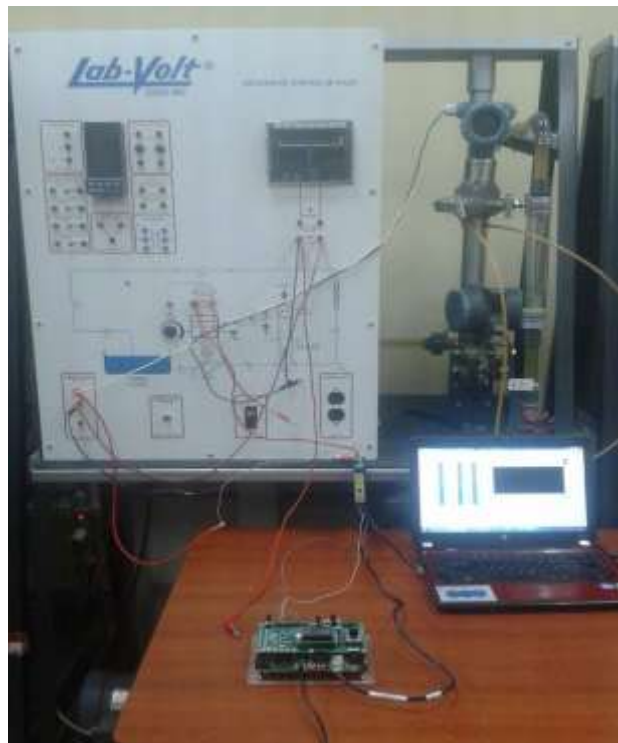


Figura 2. 33. Implementación del Control Difuso en la Planta de Flujo
Elaborado por: Alomoto, L. y Quimbita, E.

CAPÍTULO 3

DISEÑO Y CONFIGURACIÓN DE LA RED ETHERNET

3.1. DISEÑO DE LA RED ETHERNET.

El objetivo en esta parte del proyecto es la de supervisar y controlar los procesos de Presión y Flujo a través de las tarjetas SB-RIO. Para esto se diseñó una red Ethernet empleando 1 PC, 2 tarjetas SB-RIO, y un switch.

La red EtherNet/IP proporciona sistemas de red a nivel de toda la planta con el uso de tecnologías de conexión en red abiertas. Permite control información en tiempo real en aplicaciones discretas y de proceso continuo. La red EtherNet/IP conecta dispositivos tales como arrancadores de motor y sensores a controladores, dispositivos HMI, entre otros. Admite comunicaciones no industriales e industriales en una infraestructura de red común.

Al usar SB-RIO la facilidad de implementar una red aumenta gracias al software Labview que cuenta con una plataforma amigable de programación, adicionalmente facilita el diseño de HMI y SCADA en tiempo Real.

Interface Ethernet Para SB-RIO

La SB-RIO proporciona conectividad mediante el protocolo de comunicación Ethernet/IP a dispositivos con puerto serial RS232, puerto Ethernet 10/100BASE-T. [23]

Gracias a esta característica de la tarjeta SB-RIO, es posible conectar la FPGA a una red Ethernet. En la Figura 3. 1, se puede ver el puerto Ethernet que permite la conexión de una tarjeta SB-RIO a una red.



Figura 3. 1. Puerto Ethernet SB-RIO

Elaborado por: Alomoto, L. y Quimbita, E.

Hardware

Como Hardware se utilizó dos tarjetas SB-RIO que soportan comunicación Ethernet, una PC, y un switch para la integración de la red Ethernet.

Software

Cuando se trabaja con dispositivos NI es imprescindible el uso de Labview, es por eso que en la red Ethernet se usó las Shared Variable (Variables Compartidas), las mismas que pueden ser usadas con RT (Real Time).

Variable Compartida (Shared Variable)

El uso de la variable compartida, puede compartir los datos entre los bucles en un solo diagrama o entre VIs en toda la red. En contraste con

muchos métodos de intercambio de datos existentes en LabVIEW, como UDP / TCP, y FIFO en tiempo real.

Se pueden crear tres tipos de variables compartidas:

- single-process
- network-published
- time-triggered

En el diseño de la red se utilizó variables compartidas de single Process en la Figura 3. 2, se puede visualizar las propiedades de una variable compartida.

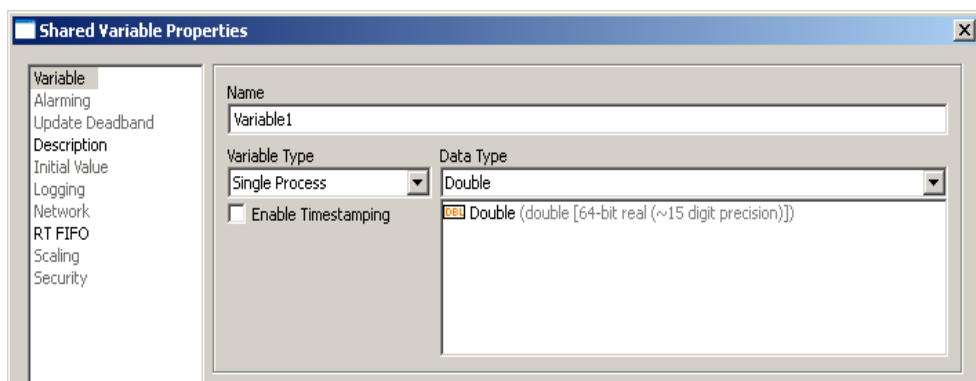


Figura 3. 2. Propiedades de una Variable Compartida

Elaborado por: Alomoto, L. y Quimbita, E.

Puede elegir entre un gran número de tipos de datos estándar para una nueva variable compartida. Además de estos tipos de datos estándar, puede especificar un tipo de datos personalizados seleccionando personalizada del tipo de datos lista desplegable y navegar a un control personalizado.

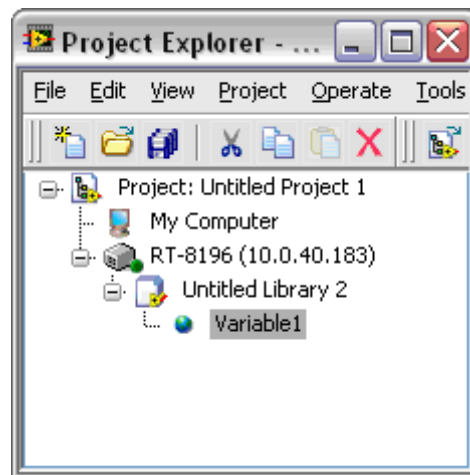


Figura 3. 3. Variable Compartida

Elaborado por: Alomoto, L. y Quimbita, E.

Después de agregar una variable compartida para un proyecto de LabVIEW, se puede colocar la variable compartida al diagrama de bloques de un VI para leer o escribir la variable compartida, como se muestra en la Figura 3. 4.

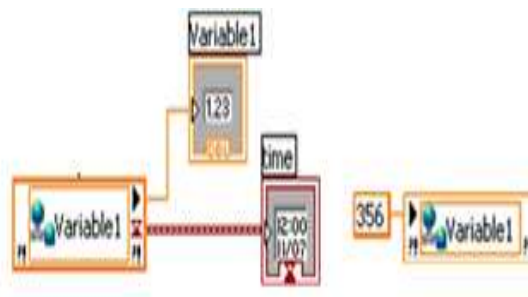


Figura 3. 4. Lectura y escritura en una variable compartida

Elaborado por: Alomoto, L. y Quimbita, E.

Variable Compartida tipo network-published

En el uso de la variable compartida tipo network-published, se puede escribir y leer variables compartidas a través de una red Ethernet. La implementación de una red es manejada en su totalidad por la variable tipo network-published. Además de hacer que los datos estén disponibles en la red, la variable compartida tipo network-published añade muchas

características que no están disponibles con la variable compartida de un solo proceso. Para proporcionar esta funcionalidad adicional, la aplicación interna de la variable compartida de red-publicado es considerablemente más complejo que el de la variable compartida de un solo proceso.

3.2. CONFIGURACIÓN DE LA RED

En la red Ethernet la PC trabajara como maestro y las tarjetas SB-RIO como esclavos.

Como primer punto seleccionamos el dominio en el que va a funcionar nuestra red, en este caso el maestro tendrá la siguiente IP 169.254.66.1, la máscara de Subred será 255.255.0.0, como se muestra en la Figura 3. 5.

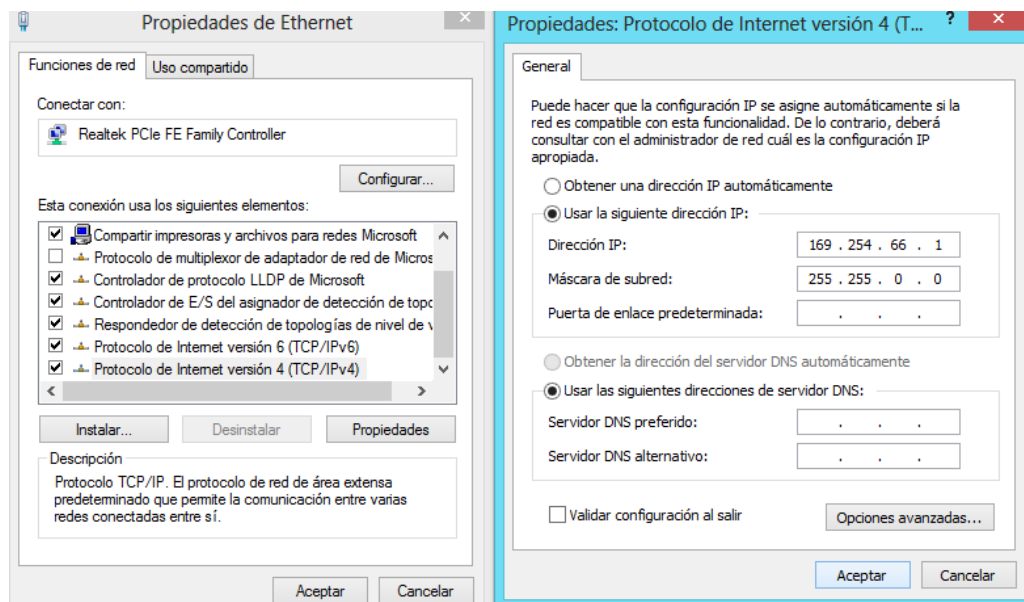


Figura 3. 5. Cambio de la Dirección IP de la PC

Elaborado por: Alomoto, L. y Quimbita, E.

Damos otra dirección IP a los esclavos, para la planta de Presión la IP es 169.254.4.42, para la planta de Flujo la IP es 169.254.66.100.

Para cambiar la IP de los esclavos se necesita abrir el software NI MAX como se muestra en la Figura 3. 6.

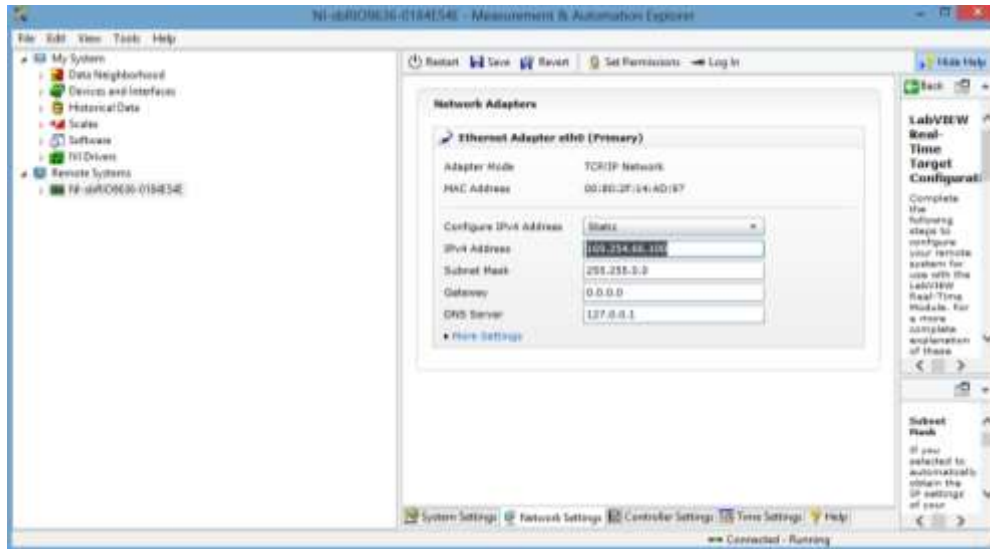


Figura 3. 6. Cambio de IP de las SB-RIO

Elaborado por: Alomoto, L. y Quimbita, E.

La máscara de Red siempre deberá ser la misma en el Maestro y en los Esclavos: 255.255.0.0, lo demás dejamos en blanco puesto que no necesitamos cambiarlo en este caso.

Después de haber establecido la IP y la máscara de Red para el Maestro y esclavos reiniciamos el equipo para evitar cualquier inconveniente de conexión a la Red.

Es conveniente que deshabilitemos el firewall de Windows que en la mayoría de los casos impide que sus recursos compartidos sean visibles a través de la Red.

Para comprobar la comunicación entre el maestro y los esclavos se recomienda hacer ping entre ellos, en la Figura 3. 7, se muestra el mensaje de una buena comunicación entre el maestro y un esclavo.

```

Administrador: Símbolo del sistema
Microsoft Windows [Versión 6.2.9200]
(c) 2012 Microsoft Corporation. Todos los derechos reservados.

C:\Windows\system32>ping 169.254.66.100

Realizando ping a 169.254.66.100 con 32 bytes de datos:
Respuesta desde 169.254.66.100: bytes=32 tiempo=1ms TTL=64
Respuesta desde 169.254.66.100: bytes=32 tiempo=1ms TTL=64
Respuesta desde 169.254.66.100: bytes=32 tiempo=1ms TTL=64
Respuesta desde 169.254.66.100: bytes=32 tiempo=1ms TTL=64

Estadísticas de ping para 169.254.66.100:
Pquetes: enviados = 4, recibidos = 4, perdidos = 0
(0% perdidos),
Tiempo aproximado de ida y vuelta en milisegundos:
Mínimo = 0ms, Máximo = 1ms, Media = 0ms

C:\Windows\system32>_

```

Figura 3. 7. Ping Maestro a Esclavo

Elaborado por: Alomoto, L. y Quimbita, E.

3.3. HMI DE LA RED ETHERNET

Para el HMI se utilizó el software Labview en el mismo se describe en forma gráfica el historial de las variables controladas cabe recalcar que este HMI se ejecuta y funciona en el maestro, es decir en la PC para posteriormente a través de la red Ethernet enviar la señal del set point (SP) hacia los controladores (FPGA's) y recibir de los mismos la variable de proceso (PV) y la Variable de control (CV).

En la Figura 3. 8, podemos visualizar el HMI de las plantas controladas



Figura 3. 8. HMI de los procesos controlados

Elaborado por: Alomoto, L. y Quimbita, E.

Finalmente en la Figura 3. 9, se puede apreciar la parte físicas del proyecto, es decir los Controlos difusos de presión y flujo utilizando FPGA's integrados por la red Ethernet.



Figura 3. 9. Controlador De Presión y Flujo integrados en una red Ethernet

Elaborado por: Alomoto, L. y Quimbita, E.

CAPÍTULO 4

PRUEBAS Y ANÁLISIS DE RESULTADO

En el presente capítulo se desarrollan las pruebas y el análisis de los resultados obtenidos de implementar los algoritmos de control difuso y control PID tradicional a las estaciones de Presión y Flujo.

4.1. PRUEBAS Y ANÁLISIS DE RESULTADO DE LOS CONTROLES IMPLEMENTADOS A LAS ESTACIONES DE PRESIÓN Y FLUJO.

4.1.1 Pruebas y análisis de resultados del control PD difusos más acción integral implementada a la estación de Presión.

Las pruebas se realizaron mediante varios cambios de Set-Point, es decir teniendo en cuenta un valor inicial y un valor final, en la Tabla 4. 1 se puede observar los resultados obtenidos, de la variable del proceso (PV), el tiempo de estabilización y el porcentaje de error existente entre la variable del proceso con respecto al punto de consigna.

El porcentaje de error que surge en cada medición, se calcula teniendo en cuenta el valor del SP y el PV mediante la Ec. 4. 1.

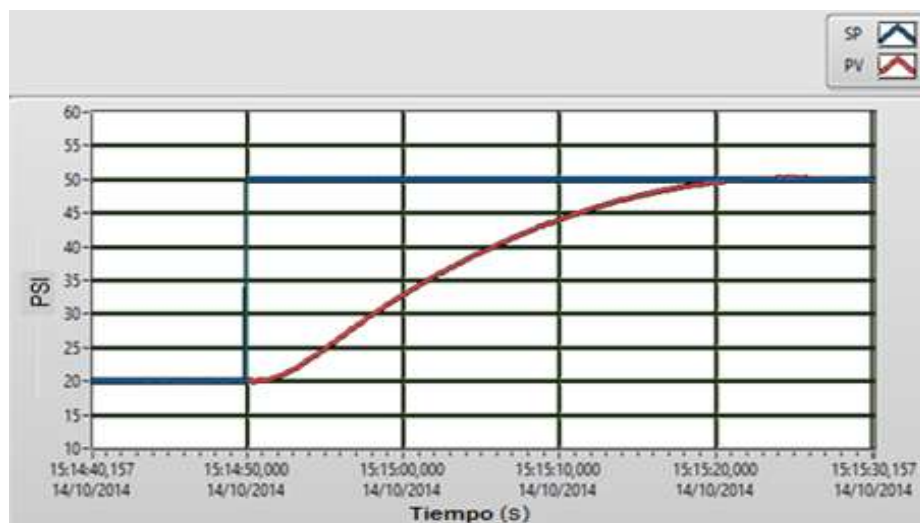
$$(\%) \text{ de Error} = \frac{SP - PV}{SP} * 100 \quad \text{Ec. 4. 1.}$$

Tabla 4. 1. Resultado obtenidos del controlador difuso.

SET-POINT INICIAL (PSI)	SET-POINT FINAL (PSI)	PV (PSI)	TIEMPO DE ESTABILIZACION (SEG)	% DE ERROR PV.vs.SP. (%)
20	50	49,9	30	0,2
50	20	20	37	0
20	45	44,9	24	0,222
45	25	25	28	0
25	35	34,9	17	0,285
PROMEDIO			27,2	0,14

Elaborado por: Alomoto, L. y Quimbita, E.

Después de varias pruebas realizadas a la estación de Presión, los resultados obtenidos son satisfactorios para este proceso, ya que como se pudo observar en la tabla anterior el promedio de error existente es mínimo, y su tiempo de estabilización es rápida, en las figuras; Figura 4. 1, Figura 4. 2, Figura 4. 3 y Figura 4. 4 se muestran los resultados reales obtenidos en la práctica, a diferentes valores de Set-Point las mismas que son ingresadas por el usuario mediante teclado.

**Figura 4. 1.** SP (20-50 PSI)

Elaborado por: Alomoto, L. y Quimbita, E.

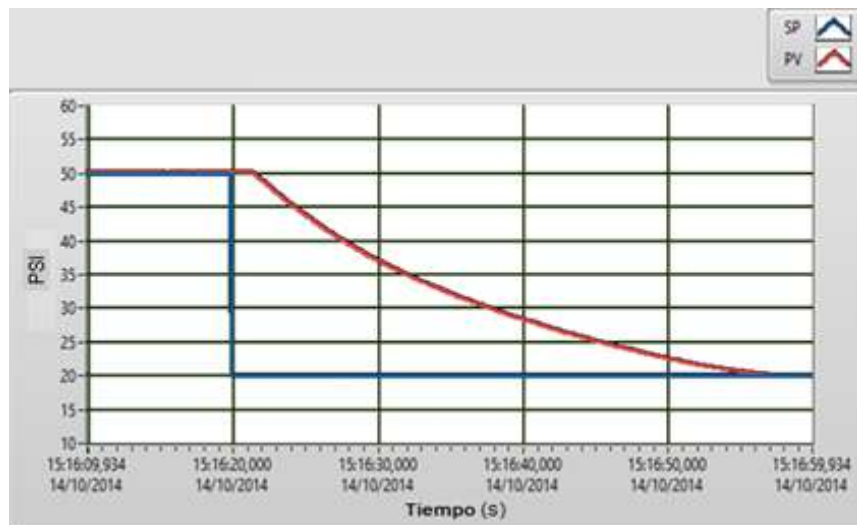


Figura 4. 2. SP (50-20 PSI)

Elaborado por: Alomoto, L. y Quimbita, E.

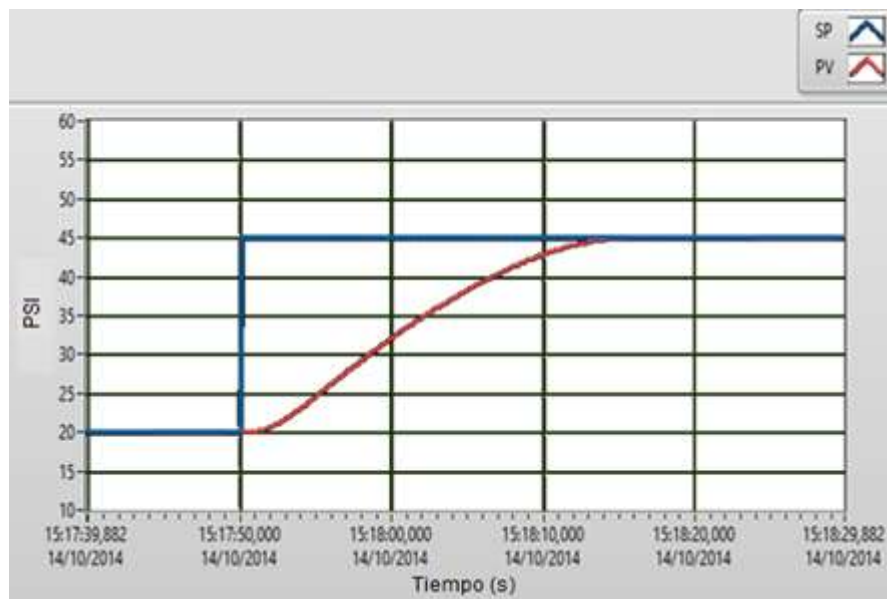


Figura 4. 3. SP (20-45 PSI)

Elaborado por: Alomoto, L. y Quimbita, E.

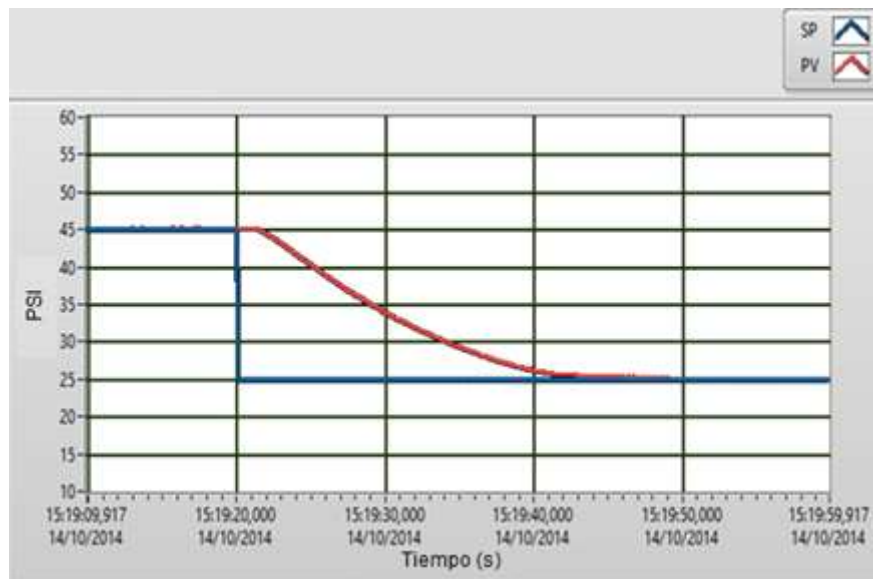


Figura 4. 4. SP (45-25 PSI)

Elaborado por: Alomoto, L. y Quimbita, E.

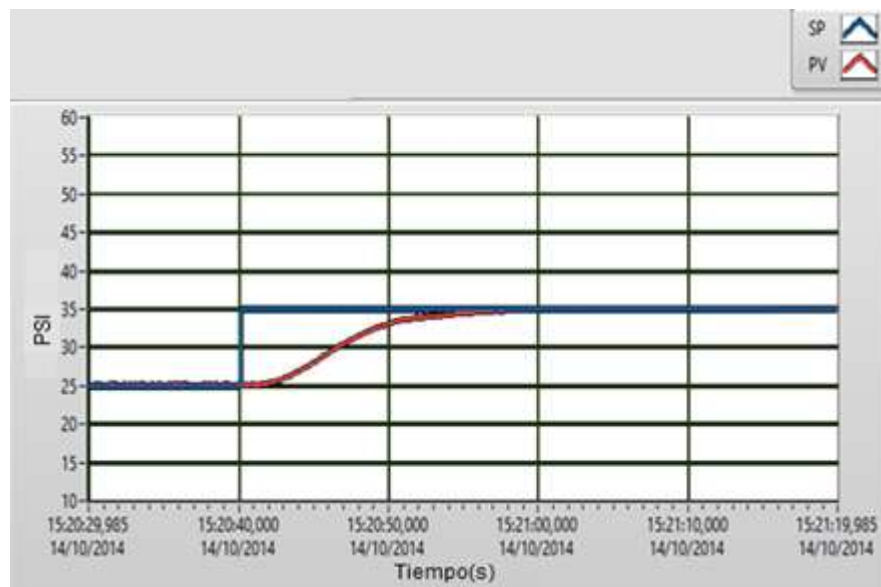


Figura 4. 5. SP (25-35 PSI)

Elaborado por: Alomoto, L. y Quimbita, E.

4.1.2 Pruebas y análisis de resultados del control PID difusos más acción integral implementada a la estación de Flujo.

Las pruebas al igual que a la estación de Presión se realizaron mediante varios cambios de Set-Point, es decir teniendo en cuenta un valor inicial y un valor final, en la Tabla 4. 2 se puede observar los resultados obtenidos, de la variable del proceso (PV), el tiempo de estabilización y el porcentaje de error existente entre la variable del proceso con respecto al punto de consigna.

Para el cálculo del error existente en las mediciones realizadas a la estación de Flujo, también se toma en consideración la Ec. 4. 1, considerada anteriormente para el cálculo del error de la estación de Presión.

Tabla 4. 2. Resultado obtenidos del controlador difuso.

SET-POINT INICIAL (inH ₂ O)	SET-POINT FINAL (inH ₂ O)	PV (inH ₂ O)	TIEMPO DE ESTABILIZACION (SEGUNDOS)	% DE ERROR (%)
12	45	45	13	0
45	18	17,9	9	0,55
18	38	37,9	10	0,26
38	28	28,2	8	0,71
45	12	11,9	8	0,83
PROMEDIO			9,6	0,47

Elaborado por: Alomoto, L. y Quimbita, E.

Después de varias pruebas realizadas a la estación de Flujo, los resultados obtenidos son satisfactorios para este proceso, como se pudo observar en la tabla anterior el promedio de error existente es de 0,47% siendo mínimo la variación entre el SP y el PV, y su tiempo de estabilización no sobrepasa los 13 seg., en todas las pruebas realizadas dando como promedio de estabilización 9,6 seg., en las figuras;

Figura 4. 6, Figura 4. 7, Figura 4. 8, Figura 4. 9 y Figura 4. 10 se muestran los resultados reales obtenidos en la práctica, a diferentes valores de Set-Point las mismas que son ingresadas por el usuario mediante teclado.

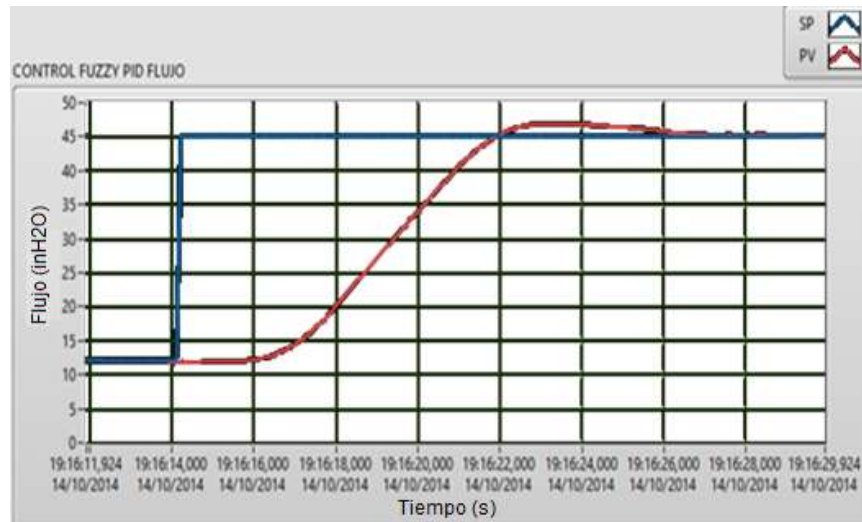


Figura 4. 6. SP (12-45 inH₂O)

Elaborado por: Alomoto, L. y Quimbita, E.

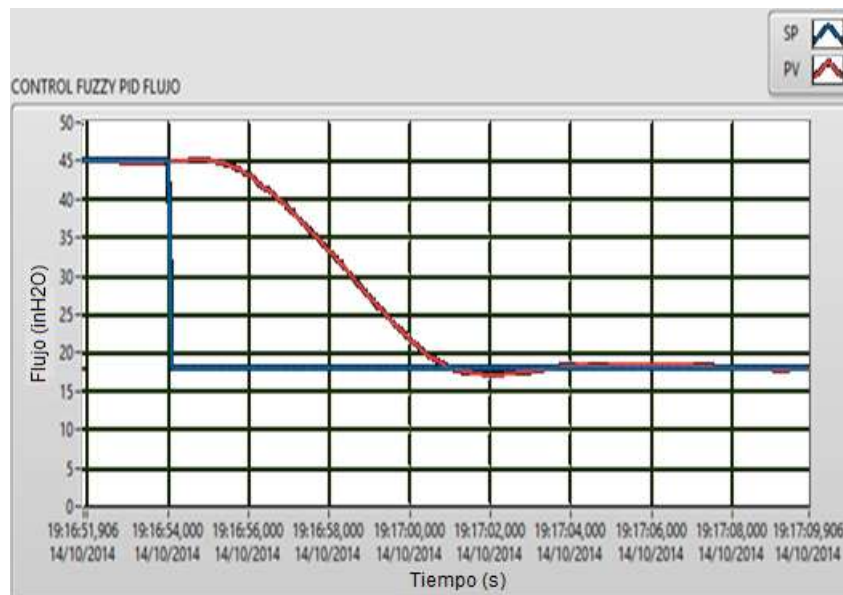


Figura 4. 7. SP (45- 18 inH₂O)

Elaborado por: Alomoto, L. y Quimbita, E.

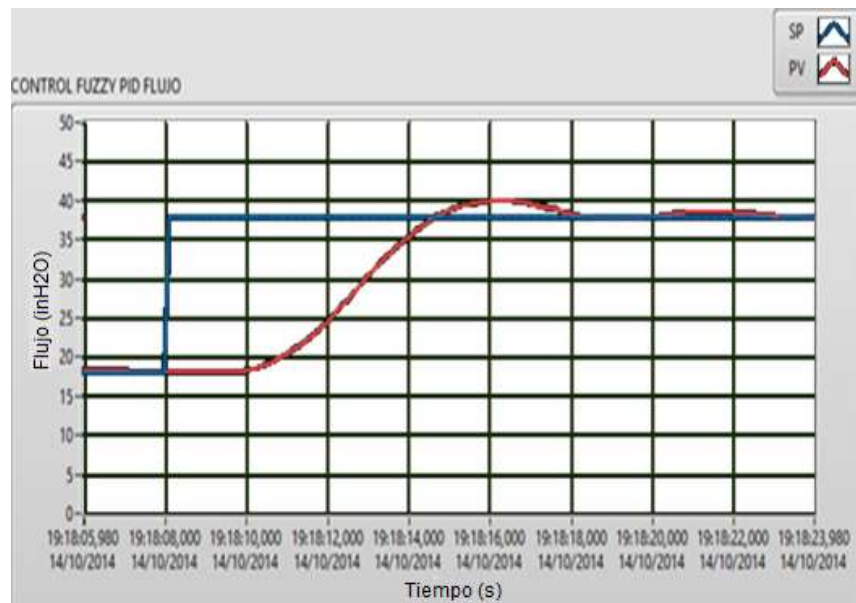


Figura 4. 8. SP (18-38 inH₂O)

Elaborado por: Alomoto, L. y Quimbita, E.

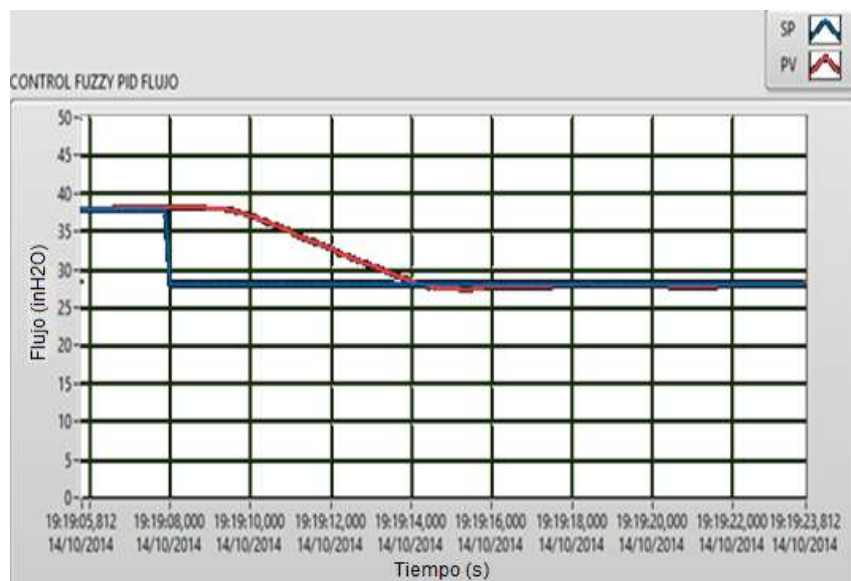


Figura 4. 9. SP (38-28 inH₂O)

Elaborado por: Alomoto, L. y Quimbita, E.

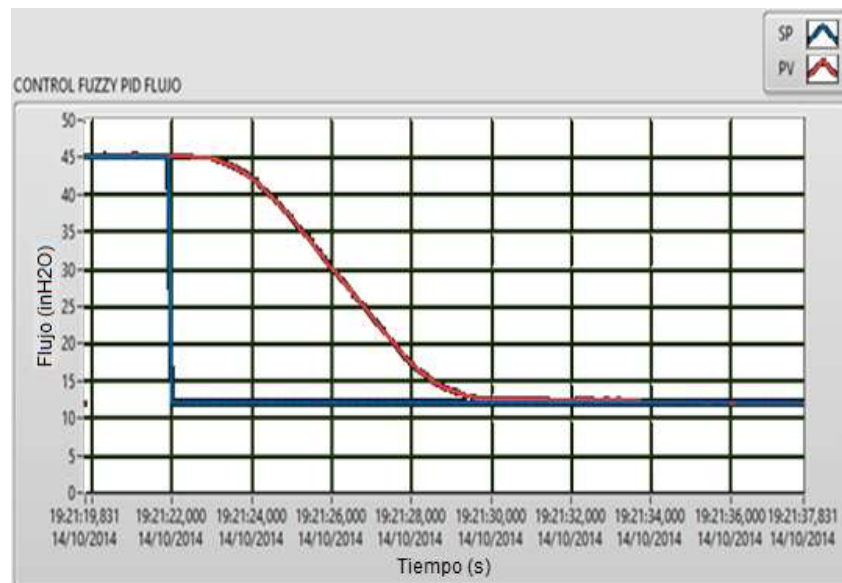


Figura 4. 10. SP (45-12 inH₂O)

Elaborado por: Alomoto, L. y Quimbita, E.

4.2. ANÁLISIS DE RESULTADO DEL CONTROLADOR AVANZADO vs. UN CONTROL TRADICIONAL IMPLEMENTADOS A LAS ESTACIONES DE PRESIÓN Y FLUJO.

4.2.1. Resultados del algoritmo PD Difuso más acción Integral vs. El control PID implementado a la estación de Presión.

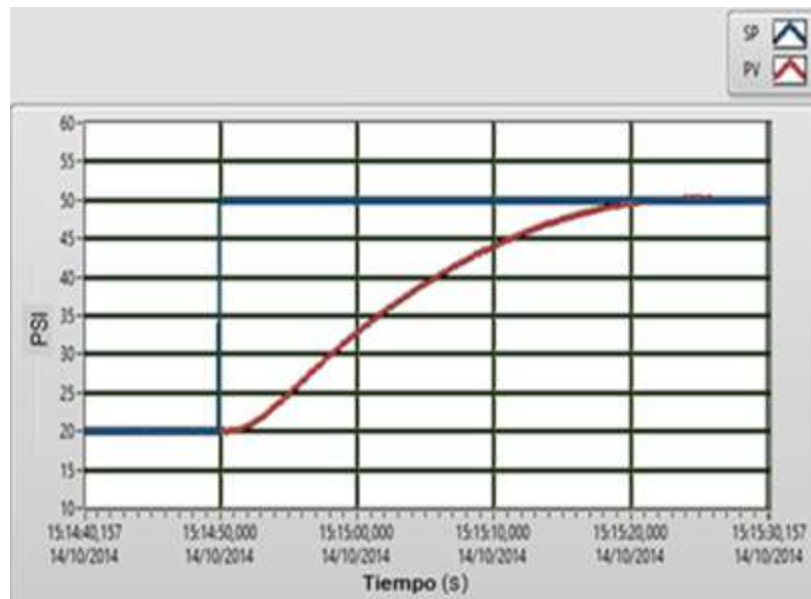
En la Tabla 4. 3, se muestran los resultados obtenidos de los dos controladores, y mediante la misma se realizó el análisis de cuál es el mejor controlador a implementarse en este tipo de procesos, se realizó la comparación de los resultados obtenidos, es decir el tiempo de estabilización y el porcentaje del sobreimpulso presente en el sistema.

Tabla 4. 3. Control PD Difuso + I vs. Control PID

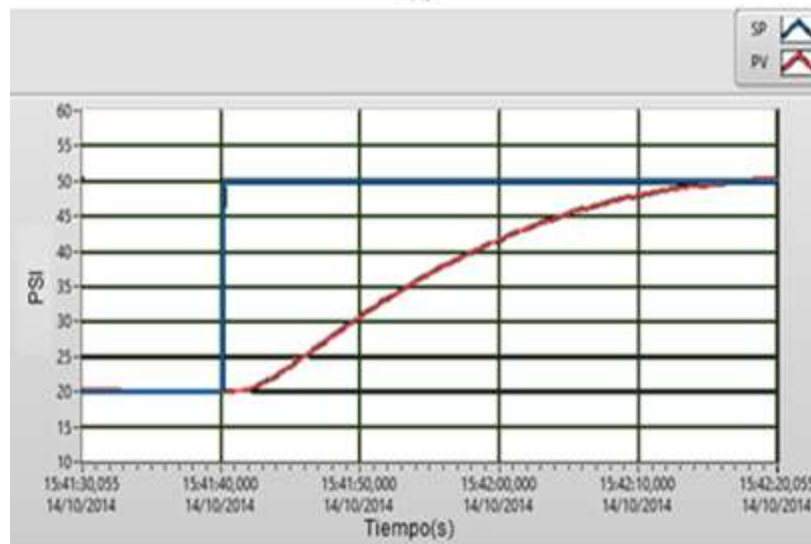
SET-POINT INICIAL (PSI)	SET-POINT FINAL (PSI)	TIEMPO DE ESTABILIZACION DEL CONTROL DIFUSO (SEG)	TIEMPO DE ESTABILIZACION DEL CONTROL PID (SEG)	% DE SOBREPULSO CONTROL DIFUSO	% DE SOBREPULSO PID
20	50	30	35	0	0
50	20	37	39	0	0
20	45	24	28	0	0
45	25	28	28	0	0
25	35	17	20	0	0
PROMEDIO		27,2	30	0	0

Elaborado por: Alomoto, L. y Quimbita, E.

En las figuras; Figura 4. 11, Figura 4. 12, Figura 4. 13, Figura 4. 14 y Figura 4. 15, se puede observar gráficamente los resultados comparativos obtenidos del control PD Difuso + I vs el control PID tradicional, a diferentes cambios de Set-Point ingresadas desde el servidor por el usuario.



(a)



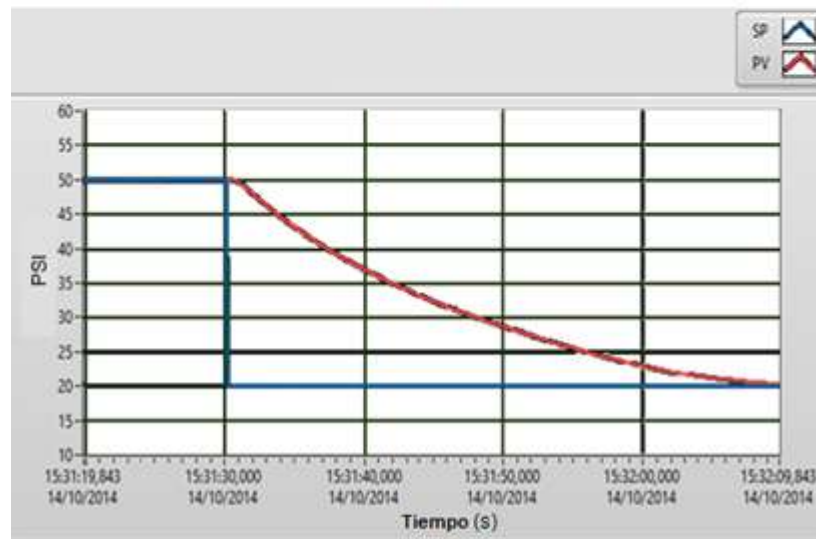
(b)

Figura 4. 11. SP 20-50 (a) Control Difuso, (b) PID.

Elaborado por: Alomoto, L. y Quimbita, E.



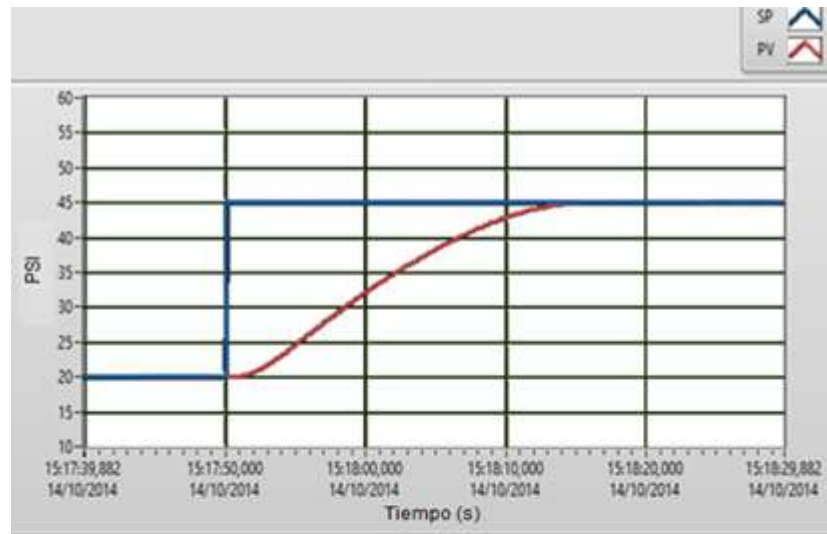
(a)



(b)

Figura 4. 12. SP 50-20 (a) Control Difuso, (b) PID.

Elaborado por: Alomoto, L. y Quimbita, E.



(a)



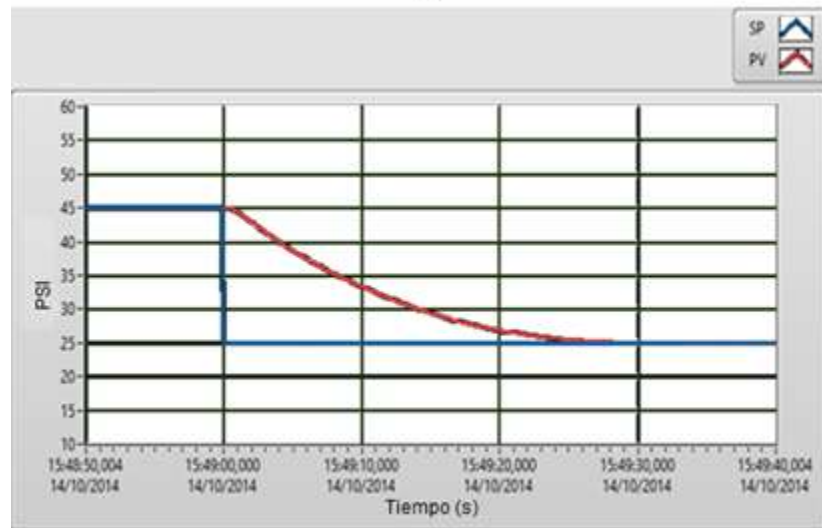
(b)

Figura 4. 13. SP 20-45 (a) Control Difuso, (b) PID.

Elaborado por: Alomoto, L. y Quimbita, E.



(a)



(b)

Figura 4. 14. SP 45-25 (a) Control Difuso, (b) PID.

Elaborado por: Alomoto, L. y Quimbita, E.



(a)

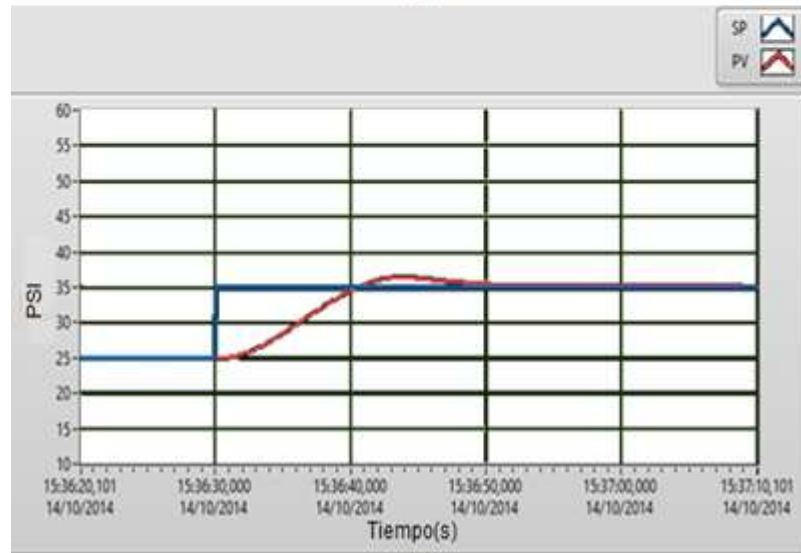


Figura 4. 15. SP 25-35 (a) Control Difuso, (b) PID.

Elaborado por: Alomoto, L. y Quimbita, E.

Análisis de resultado del algoritmo PD Difuso + I vs. Controlador PID Implementado en la estación de Presión.

De acuerdo a los resultados obtenidos, el controlador difuso tiende a estabilizarse con mayor rapidez que el controlador PID tradicional en la Figura 4. 16, podemos observar que el control difuso es quien tiene resultados favorables y por ende es factible su implementación en este tipo de proceso.

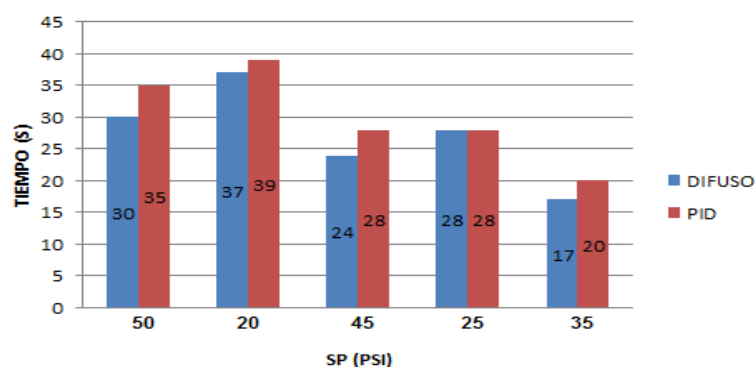


Figura 4. 16. Tiempo de estabilización de los controladores

Elaborado por: Alomoto, L. y Quimbita, E.

En la Figura 4. 17, se puede observar los resultados del sobreimpulso, los cuales no se presentan con ninguno de los dos controladores.

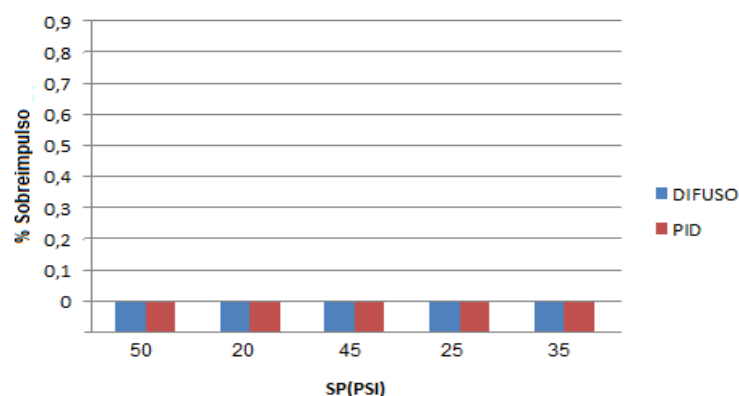


Figura 4. 17. Porcentaje de sobreimpulso de los controladores.

Elaborado por: Alomoto, L. y Quimbita, E.

4.2.2. Resultado del algoritmo PID Difuso más acción Integral vs. El control PID implementado a la estación de Flujo.

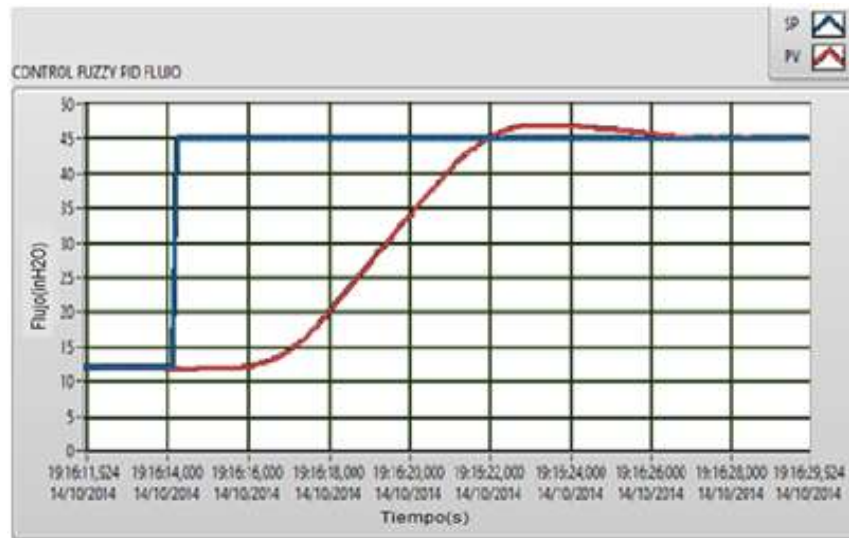
En la Tabla 4. 4, se muestran los resultados obtenidos de los dos controladores, y mediante la misma se realiza el análisis de cuál es el mejor controlador a implementarse en este tipo de procesos, se realizó la comparación de los resultados obtenidos, es decir el tiempo de estabilización y el porcentaje del sobre-impulso presente en el sistema.

Tabla 4. 4. Control PID Difuso + I vs. Control PID

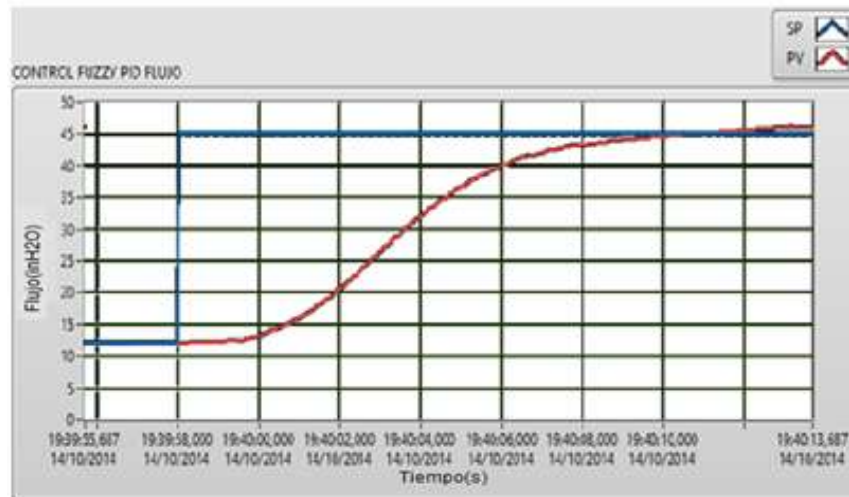
SET-POINT INICIAL (inH2O)	SET-POINT FINAL (inH2O)	TIEMPO DE ESTABILIZACION DEL CONTROL DIFUSO (SEG)	TIEMPO DE ESTABILIZACION DEL CONTROL PID (SEG)	% DE SOBREPULSO DEL CONTROL DIFUSO (%)	% DE SOBREPULSO DEL PID (%)
12	45	13	17	4,4	2,2
45	18	9	20	5,5	0
18	38	10	12	5,2	0
38	28	8	11	1,7	0
45	12	8	25	0	0
PROMEDIO		9,6	17	3.36	0,44

Elaborado por: Alomoto, L. y Quimbita, E.

En las figuras; Figura 4. 18, Figura 4. 19, Figura 4. 20, Figura 4. 21 y Figura 4. 22, se puede observar gráficamente los resultados comparativos obtenidos del control PID Difuso + I vs el control PID tradicional, a diferentes cambios de Set-Point ingresadas desde el servidor por el usuario.



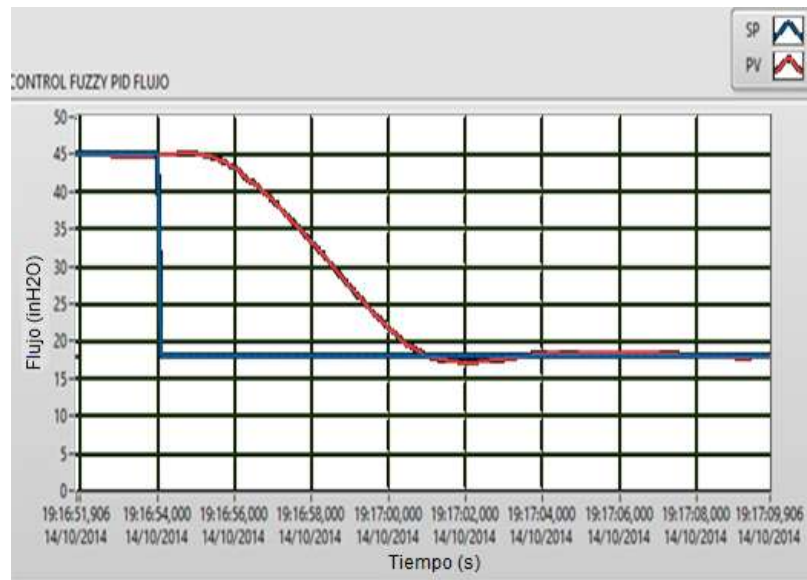
(a)



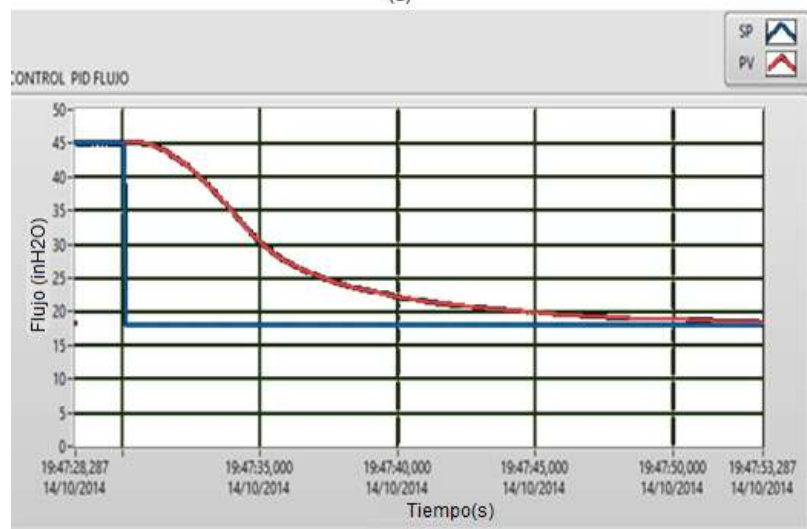
(b)

Figura 4. 18. SP 12-45 (a) Control Difuso, (b) PID.

Elaborado por: Alomoto, L. y Quimbita, E.



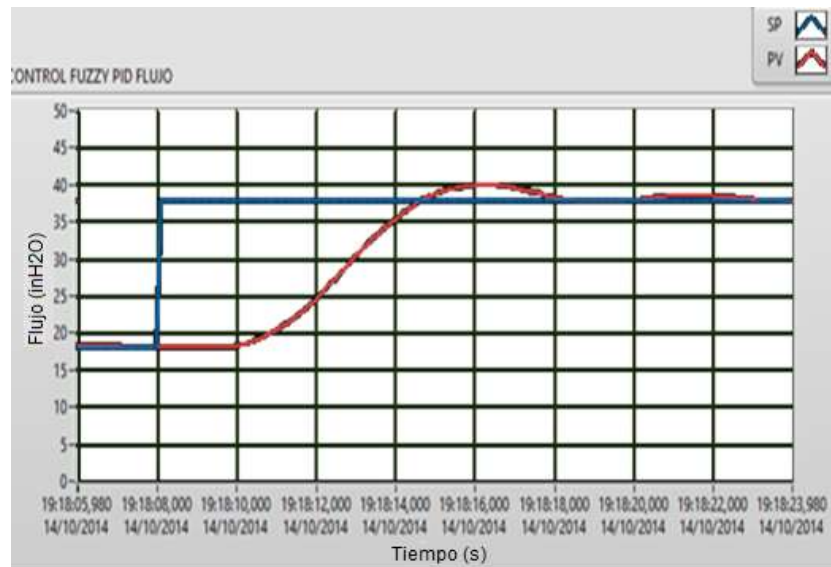
(a)



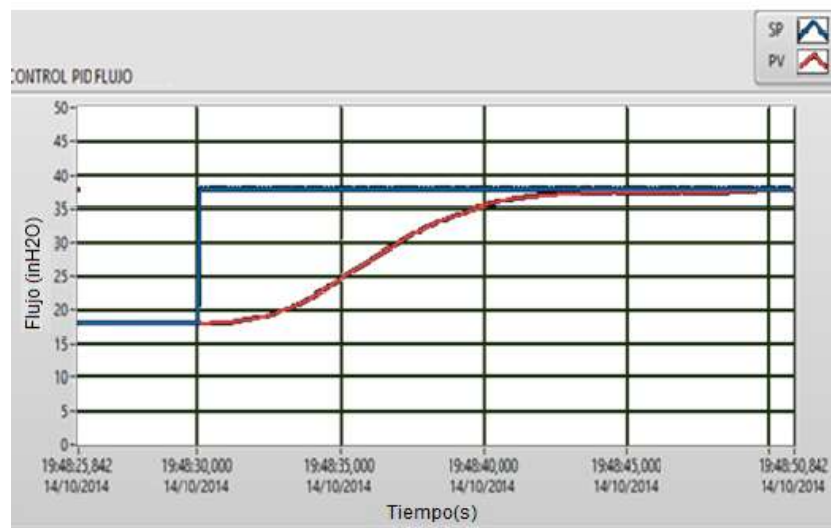
(b)

Figura 4. 19. SP 45-18 (a) Control Difuso, (b) PID.

Elaborado por: Alomoto, L. y Quimbita, E.



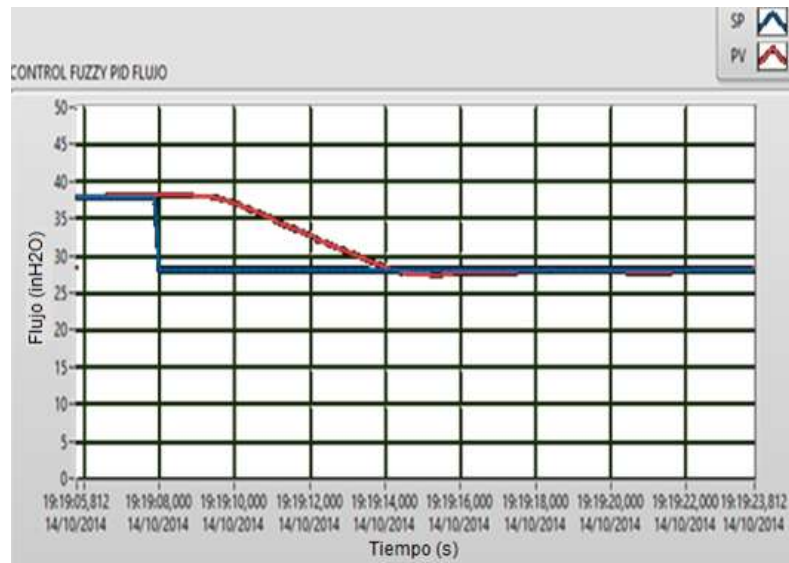
(a)



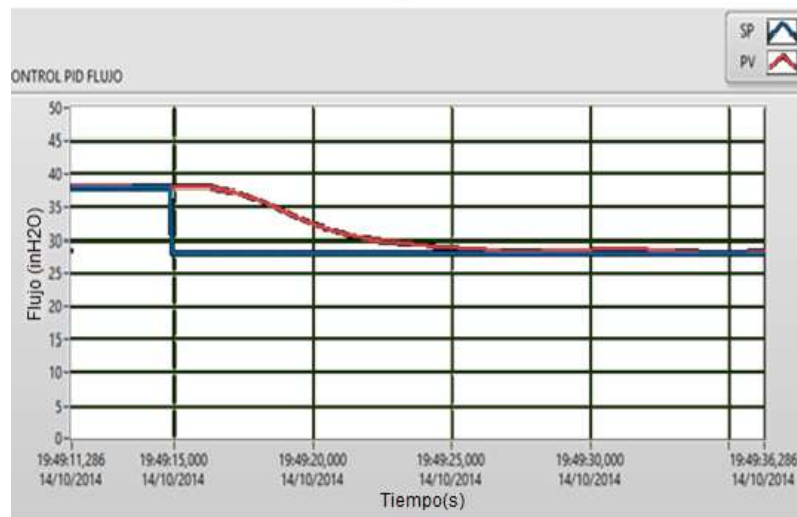
(b)

Figura 4. 20. SP 18-38 (a) Control Difuso, (b) PID.

Elaborado por: Alomoto, L. y Quimbita, E.



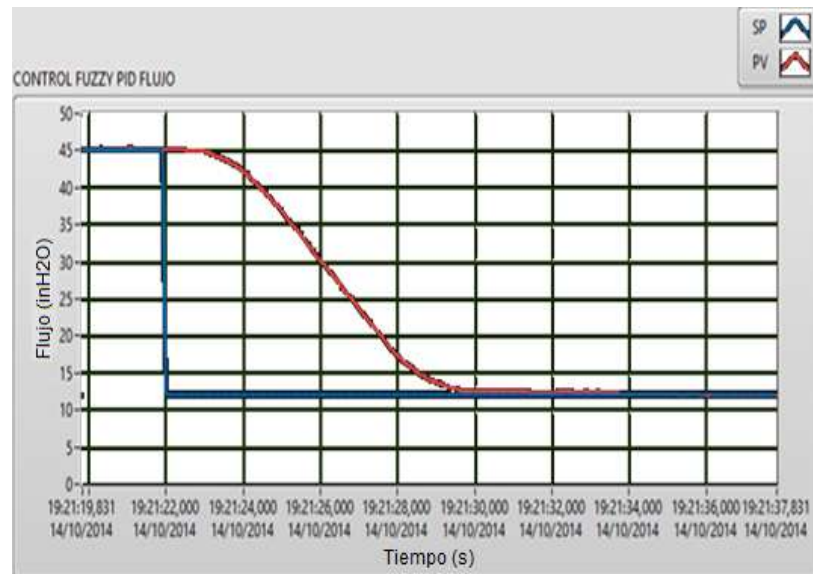
(a)



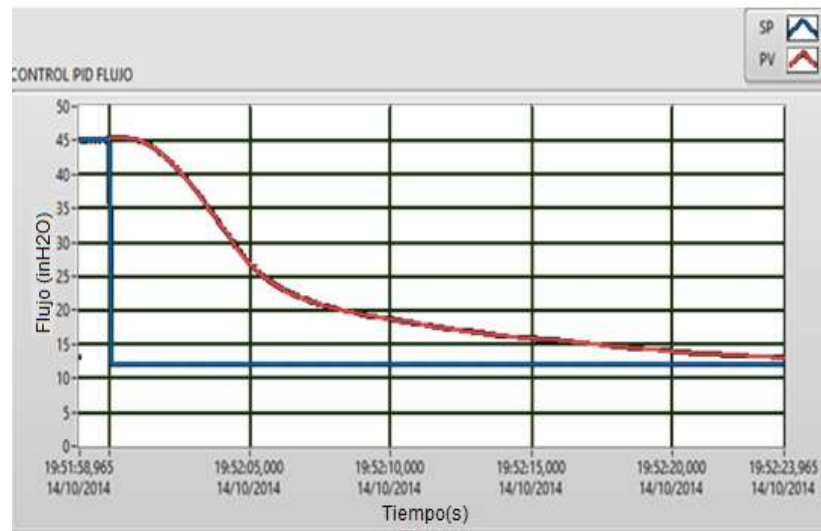
(b)

Figura 4. 21. SP 38-28 (a) Control Difuso, (b) PID.

Elaborado por: Alomoto, L. y Quimbita, E.



(a)



(b)

Figura 4. 22. SP 45-12 (a) Control Difuso, (b) PID.

Elaborado por: Alomoto, L. y Quimbita, E.

Análisis de resultado del algoritmo PID Difuso + I vs. Controlador PID Implementado en la estación de Flujo.

Con los resultados obtenidos de las pruebas realizadas con los dos controladores, se puede observar que el controlador difuso tarda notablemente menos en estabilizarse en un proceso rápido en comparación al controlador PID tradicional como se muestra en la Figura 4. 23.

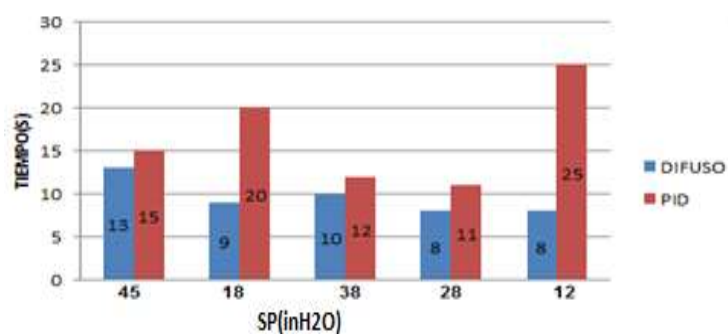


Figura 4. 23. Tiempo de estabilización de los controladores

Elaborado por: Alomoto, L. y Quimbita, E.

En la Figura 4. 24 se puede observar los resultados obtenidos al implementar el control Difuso y el control PID a la estación de Flujo, se puede ver que en procesos rápidos se hace presente el sobreimpulso la misma que no afecta en la estabilización del proceso y por ende es factible la implementación del control PID Difuso + I en esta estación.

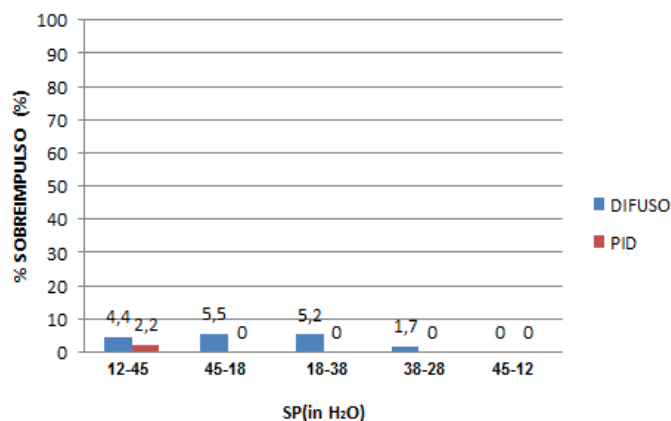


Figura 4. 24. Porcentaje de sobreimpulso

Elaborado por: Alomoto, L. y Quimbita, E.

4.3. PRUEBAS DE FUNCIONAMIENTO DE LA RED ETHERNET

La PC (maestro) y los dispositivos SB RIO (esclavos) deben tener una dirección IP similar, para la configuración del maestro (PC) se debe ingresar al protocolos de internet versión 4 (TCP/IPv4), como se observa en la Figura 4. 25, y se modifica la IP.



Figura 4. 25. Dirección IP de la PC

Elaborado por: Alomoto, L. y Quimbita, E.

Para configurar la IP de los dispositivos SB RIO, es necesario dirigirse al NI MAX y modificar la IP como se muestra en la Figura 4. 26.

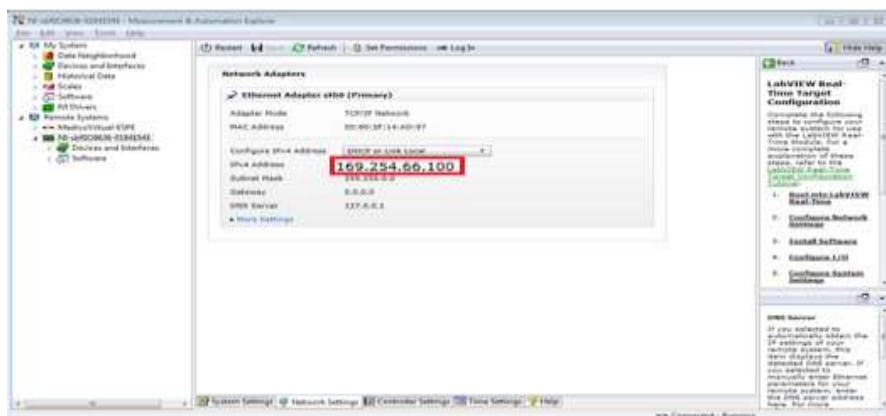
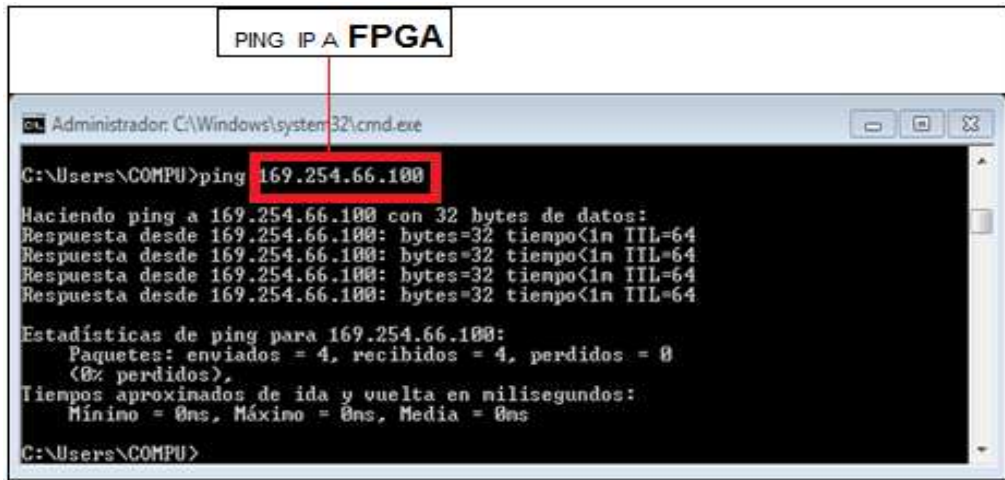


Figura 4. 26. Dirección IP del Dispositivo SB RIO

Elaborado por: Alomoto, L. y Quimbita, E.

Con ayuda del interpretador de comandos CMD se debe comprobar la conexión de red haciendo PING a la dirección IP de cada dispositivo,

con la finalidad de comprobar que existe comunicación, con él envió (4) y la recepción (4) de un paquete de datos, si todo está correcto los datos perdidos será cero (0) como se observa en la Figura 4. 27.



```

PING IP A FPGA
Administrador: C:\Windows\system32\cmd.exe
C:\Users\COMPU>ping 169.254.66.100
Haciendo ping a 169.254.66.100 con 32 bytes de datos:
Respuesta desde 169.254.66.100: bytes=32 tiempo<1n TTL=64
Respuesta desde 169.254.66.100: bytes=32 tiempo<1n TTL=64
Respuesta desde 169.254.66.100: bytes=32 tiempo<1n TTL=64
Respuesta desde 169.254.66.100: bytes=32 tiempo<1n TTL=64
Estadísticas de ping para 169.254.66.100:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (<0% perdidos>),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 0ms, Máximo = 0ms, Media = 0ms
C:\Users\COMPU>
  
```

Figura 4. 27. Envío y recepción de datos en CMD

Elaborado por: Alomoto, L. y Quimbita, E.

Para comprobar el envío y la recepción de los datos se utilizó “Shared variable” como se observa en la Figura 4. 28, se realiza el envío de una señal de SP (Set-Point) al cliente a través de la Shared Variable Set-Point configurado en modo de escritura, y este a su vez recibe la señal mediante la misma Shared variable, pero ahora configurado en modo de lectura como se observa en la Figura 4. 29.

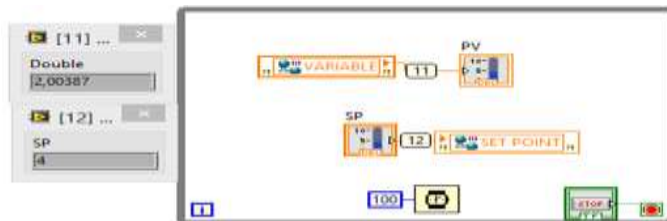


Figura 4. 28. VI de la PC Servidor

Elaborado por: Alomoto, L. y Quimbita, E.

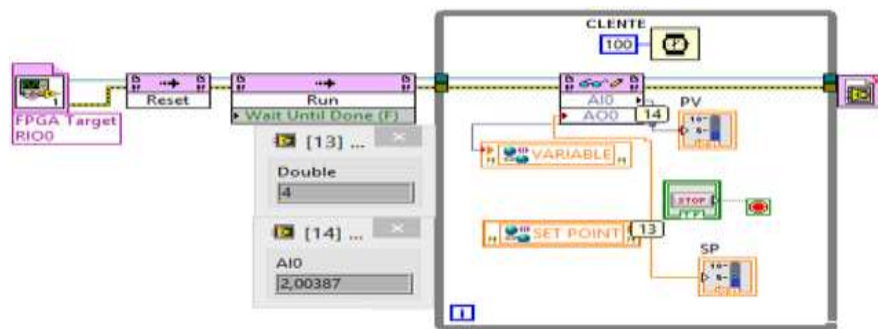


Figura 4. 29. VI Cliente

Elaborado por: Alomoto, L. y Quimbita, E.

En la Figura 4. 30, podemos observar la transmisión y la recepción de los datos enviados desde el servidor (PC) hacia los clientes y viceversa, justificando así que es posible realizar una red con este tipo de controladores.

Probe(s)	Value
Servidor.vi (Pro)	
[11] Double	2.004E+0
[12] SP	4.000E+0
Cliente.vi (Proy)	
[13] Double	4.000E+0
[14] AIO	2.004E+0

Figura 4. 30 Ventana de visualización de señales

Elaborado por: Alomoto, L. y Quimbita, E.

CAPÍTULO 5

CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones.

1. El controlador Difuso alcanza un control superior a un controlador tradicional como es el PID y es importante tomar en consideración el proceso ya que dependerá de ella que tan óptimo es el control, un controlador Difuso puede llegar a ser una excelente opción para suplantar un control tradicional.
2. Es posible el diseño e implementación de un controlador Difuso, mediante un dispositivo de control avanzado, para el control de las estaciones de Presión y Flujo del Laboratorio de Redes Industriales y Control de Procesos de la Universidad de las Fuerzas Armadas-ESPE integrados por una red Ethernet.
3. Al implementar un Controlador PD Difuso (Proporcional derivativo) a un proceso (estación de Presión), es posible eliminar el error en estado estable, algo que sería imposible obtener con un Control Proporcional Derivativo tradicional, obligatoriamente necesitaría de una parte integral que ayude a eliminar este error.
4. La gran ventaja que se consigue de un Control Difuso en comparación al Control PID tradicional implementado a los procesos, es el tiempo de respuesta que se obtuvo, este tiempo es mucho menor ocasionando que el control de la planta se estabilice con mayor rapidez que el control PID.
5. Después de varias pruebas realizadas, los dos controles presentan valores muy pequeños de error en estado estable, sin embargo con

el Control Difuso se obtiene un error más pequeño que el control PID tradicional, por tal razón se consigue que las variaciones del control de los procesos en estado estable sean mínimas y alcancen una variación imperceptible.

6. El Controlador Difuso responde mejor y más rápido a las perturbaciones y cambios de SP que se presentan en el transcurso de control del proceso, mientras que un PID tradicional es más lento en su respuesta.
7. La red Ethernet implementada es factible con estos dispositivos de control avanzado, debido que es posible el intercambio (transmisión y recepción) de datos en tiempo real.
8. La implementación del controlador Difuso en los dispositivos SB RIO son óptimos, ya que se puede realizar la transmisión y la recepción de los datos en tiempo real, los mismos que son monitoreados en una PC maestra conectadas mediante una red Ethernet.

5.2 Recomendaciones

1. Utilizar el control de lógica difusa cuando no es necesario contar con el modelo matemático de la planta, para realizar el control de la misma.
2. Utilizar funciones de membresía tipo triangular y trapezoidal en controles de lógica difusa, ya que sus cálculos matemáticos son sencillos y a su vez cuando sea necesario una respuesta de control rápida en el proceso.
3. Utilizar la FAM (matriz de reglas difusas), para definir las reglas que tendrá el controlador difuso con respecto a las variables de entrada y salida.
4. Implementar el control Difuso para procesos los cuales requieran de un tiempo de estabilización rápida.
5. Implementar el control difuso para procesos en los cuales haya alta presencia de ruido.

REFERENCIAS BIBLIOGRÁFICAS

BIBLIOGRAFÍA

- [1] L. A. S. Dediós, "Control predictivo y optimización económica aplicados", Piura, 2004.
- [2] S. Joe Qin y Thomas A, "Badgwell An Overview of Industrial Model Predictive Control Technology", 2000.
- [3] D. E. Rivera, "PID Controller Tuning using Internal Model Control", A Modern Approach, 2002.
- [4] Greg Goebel, "An Introduction To Fuzzy Control Systems".
- [5] Keith Sevcik, "An introduction to Model Reference Adaptive Control (MRAC)", 2006.
- [6] Bonifacio Martínez del Brío, Alfredo Sanz Molina, "Redes Neuronales y Sistemas Borrosos", Alfaomega, Tercera edición, 2007.
- [7] Pineda Cando Marco Antonio, Vivas Paspuel Atal Kumar, "Control de un modelo aerodinámico aplicando sistemas difusos", Escuela Politécnica Nacional de Quito, Tesis de Ingeniería, Noviembre 2008.
- [8] Kevin M. Passino & Stephen Yurkovich, "Fuzzy Control", Addison Wesley, 1998.
- [9] Jiménez L., "Algunas aplicaciones de la lógica difusa a teoría de control", Instituto Politécnico.

- [10] Timothy J. Ross, "Fuzzy Logic with engineering applications", John Wiley & Sons Ltd, 2004.
- [11] Ahmad M. Ibrahim, "FUZZY LOGIC for Embedded System Applications", Newnes, 2004.
- [12] Ponce Pedro, "Inteligencia Artificial con aplicaciones a la ingeniería", Alfaomega, Primera edición, 2010.

LINKOGRAFÍA

- [13] Características de una red Ethernet Industrial [Online].
<http://www.emb.cl/electroindustria/articulo.mvc?xid=1034&edi=55>
- [14] NI CompactRIO. [Online].
<http://www.ni.com/compactrio/whatis/esa/>
- [15] NI myRIO. [Online].
<http://sine.ni.com/nips/cds/view/p/lang/es/nid/211736>
- [16] NI Single-Board RIO. [online].
<http://www.ni.com/singleboard/whatis/esa/>
- [17] NI LabVIEW RIO Evaluation KIT. [online]
<http://sine.ni.com/ds/app/doc/p/id/ds-468/lang/es>
- [18] Software de Desarrollo de Sistemas NI LabVIEW. [online].
<http://www.ni.com/labview/esa/>
- [19] Module RT y FPGA. [Online].
<http://sine.ni.com/nips/cds/view/p/lang/es/nid/11834>

- [20] Tesis: “Diseño e implementación de un sistema SCADA utilizando Factorytalk y Ethernet Industrial con tecnología Allen Bradley para el monitoreo y control de las estaciones de flujo, nivel y presión en el Laboratorio de Redes Industriales y Control de Procesos de la Universidad de las Fuerzas Armadas ESPE Extensión Latacunga.” [Online].
<http://repositorio.espe.edu.ec/handle/21000/2712/browse?type=author&order=ASC&rpp=20&value=Pullupaxi+Galora%2C+Javier+Israel>
- [21] MATLAB. [Online].
http://www.mathworks.es/products/matlab/index.html?s_tid=gn_loc_drop
- [22] SISTEMA DE IDENTIFICACION. [Online].
<http://www.mathworks.es/es/products/sysid/index.html>
- [23] Control Mediante Lógica Difusa
<http://www2.elo.utfsm.cl/~elo377/documentos/Fuzzy.pdf>